

现代应用数学手册

《现代应用数学手册》编委会

计算与数值分析卷

清华大学出版社

现代应用数学手册

《现代应用数学手册》编委会

计算与数值分析卷

清华大学出版社
北京

内 容 简 介

本书是进行科学计算的常备工具书，内容新颖，查阅方便，实用性强. 主要介绍生产、科研、管理、教学等实践中在计算机上使用的各种计算方法和技巧. 全书分为14章，依次为数值计算概论、插值法、函数逼近与曲线拟合、数值积分与数值微分、方程求根、线性方程组的直接解法和迭代解法、矩阵特征值问题、非线性方程组数值解与最优化方法、常微分方程初值问题和边值问题的数值解法、偏微分方程的数值解法、多重网格法和积分方程数值解法. 每种方法均配有例题，便于读者理解、掌握和使用. 书末还附有中文-外文索引、外文-中文索引以及外国人名表.

本书可供广大科研人员、技术人员、管理干部、计算工作者及高等院校师生使用.

图书在版编目(CIP)数据

现代应用数学手册. 计算与数值分析卷/《现代应用数学手册》编委会编.
—北京：清华大学出版社，2005.1
ISBN 7-302-09831-X

Ⅰ. 现…　Ⅱ. 现…　Ⅲ. ①应用数学—手册 ②计算方法—手册 ③数值计算—手册　Ⅳ. O29-62

中国版本图书馆 CIP 数据核字(2004)第111821号

出 版 者：清华大学出版社　　地　址：北京清华大学学研大厦
http://www.tup.com.cn　　邮　编：100084
社 总 机：010-62770175　　客户服务：010-62776969
组稿编辑：刘　颖
文稿编辑：王海燕
印 刷 者：清华大学印刷厂
装 订 者：北京国马印刷厂
发 行 者：新华书店总店北京发行所
开　本：140×203　印张：27.125　字数：680千字
版　次：2005年1月第1版　2005年10月第2次印刷
书　号：ISBN 7-302-09831-X/O·420
印　数：3001～5000
定　价：48.00元

《现代应用数学手册》
编辑委员会



计算与数值分析卷

责任编委　顾丽珍

章次	编　者	校　者
1～2	陈景良　李庆扬	陆金甫
3	李庆扬	陆金甫
4	陆金甫	李庆扬
5	李庆扬	陆金甫
6	陈景良　顾丽珍	李庆扬
7	顾丽珍	李庆扬
8	顾丽珍	陆金甫
9	李庆扬	关　治
10～11	关　治	顾丽珍
12	陆金甫　关　治	关　治　顾丽珍
13	顾丽珍	李庆扬
14	陆金甫	关　治

序

随着计算机科学技术的飞速发展，人类正进入信息时代.

信息时代是应用数学大发展的时代，人类长期积累起来的知识体系，正面临着第3次数学化.数学思想，数学方法与数学模型随着计算机的广泛应用，日益渗透到各种行业中去.

当代，除了古典的数学理论（初等数学，微积分学，微分方程，复变函数等）早已得到广泛的应用外，一些比较抽象的现代数学理论（集合论、数理逻辑、范畴论、抽象代数、泛代数、代数几何、拓扑学、泛函分析等）以及一些新兴的数学理论（随机过程、时间序列、运筹学、最优化理论、有限元方法、模糊数学、混沌与分形等）也逐渐地成为社会生产，科学实验，工程技术及经济管理中不可缺少的工具，应用数学的适用范围正在迅速地扩大.

为了满足日益增长的社会需求，清华大学应用数学系《现代应用数学手册》编委会，组织编写了这套多卷集的手册.

本书读者是理、工、医、农、经管等各个领域中的广大工程技术人员、科研人员，大、中专院校的教师、学生、研究生及其他使用数学工具的实际工作者.其中有些内容对于中学生也是适用的.

编者力求使本书成为一套高质量的工具书，它有下列特点：

(1) **内容"新颖"** 本书力求做到内容现代化，除用现代观点介绍古典内容外，对已出现的新理论、新方法尽量优先选入.

(2) **突出"应用"** 本书在选材上突出数学理论的应用，以通俗易懂的方式着重介绍在现代科学技术等实际领域中应用广泛的

数学理论和方法.

(3) **紧密"结合"计算机应用** 为了更有效地应用数学方法解决各种实际问题,广大科技人员迫切要求数学方法与计算机应用相结合,提高工作效率.为此,本书在结合计算机应用方面,给予特别的重视.

(4) **版面设计"合理",便于迅速查阅** 为方便读者使用,本书采用了一套较为完善的索引体系.除正文中章、节的编号沿用国际通行的十进制编号外,对于重要的定义、定理、例题、公式、图、表等均有编号.读者可以从(1) 目录,(2) 中文—外文索引,(3) 外文—中文索引等三种途径,迅速找到所需资料.此外,本书对载人的外国科学家人名,尽量采用"名从主人"的原则.

(5) **数学符号力求"统一"与国际化** 鉴于目前国内各种文献、书籍中使用的数学符号不够统一与国际化,增加了读者阅读时的困难.本书除按国家标准 GB 3102—93 外,兼用国际数学界权威著作《数学大百科辞典》(Encyclopedic Dictionary of Mathematics, EDM)中的符号为标准.对于不在上述文献中的其他新符号,则选用较为流行者.

本手册各卷内容独立完整,便于个人读者与团体读者按需选购.当前应用数学急剧发展,编委会在条件成熟的时候,还将增出新卷.

本书的编撰是与清华大学应用数学系领导,特别是萧树铁教授的热心支持,编辑委员会各位编委的通力协作,校内外的许多教师、科研工作者的大力支持分不开的,编者深致谢意.

在编辑出版过程中,还得到清华大学出版社的热情支持.

本书从编撰到出版,历尽艰辛.饮水思源,编者还要感谢本书的发起人,清华大学应用数学系陆璇教授,北京出版社李利军编辑

及已故的北京出版社社长王政人先生.

最后,编者还要对夫人王华敏表示谢忱,没有她的深刻理解、热情支持与持久的帮助,本书也难以问世.

主编　马振华

1997年于清华园



数学符号表

$\overset{\text{def}}{=\!=}$	定义
$:=$	定义且赋值/赋值
$\operatorname{sgn}a$	符号函数 $\operatorname{sgn}a=\begin{cases}1, & a\geqslant 0,\\ -1, & a<0\end{cases}$
$\binom{a}{n}$	即 $\dfrac{a(a-1)\cdots(a-n+1)}{n!}$ （a 为实数）
$\binom{n}{k}/\mathrm{C}_n^k$	二项式系数，即 $\dfrac{n(n-1)\cdots(n-k+1)}{k!}$
$\sum\limits_{i=1}^{n}$	对 i 从 1 到 n 项求和
$\prod\limits_{i=1}^{n}$	对 i 从 1 到 n 项求积
i	虚数单位，$\mathrm{i}=\sqrt{-1}$
$\operatorname{Re}z$	复数 z 的实部
$\operatorname{Im}z$	复数 z 的虚部
$\lvert z\rvert$	复数 z 的模
$\arg z$	复数 z 的复角
$\bar{z}$	复数 z 的共轭
(a,b)	开区间
$[a,b]$	闭区间
$(a,b],[a,b)$	半开区间
∞	无穷大
$\to$	收敛于/映射

$\Leftrightarrow$	等价
$\Rightarrow$	推理
$\lim a_n$	序列$\{a_n\}$的极限
$f(x)$	函数 f 在点 x 的值
$f/f(\cdot)$	函数
$f[\cdot]$	函数 f 的均差
$\frac{\mathrm{d}f}{\mathrm{d}x}\bigg/f'$	函数 f 对 x 的导数
$\frac{\mathrm{d}^n f}{\mathrm{d}x^n}\bigg/f^{(n)}$	函数 f 对 x 的 n 阶导数
$\frac{\partial f}{\partial x}\bigg/f_x$	函数 f 对 x 的偏导数
$\frac{\partial^2 f}{\partial x\partial y}\bigg/f_{xy}$	函数 f 对 x,y 的混合偏导数
$\frac{\partial^{n+m} f}{\partial x^m\partial y^n}\bigg/f_{x^m y^n}$	函数 f 先对 x 求 m 次偏导数,再对 y 求 n 次偏导数
$\tilde{f}(x)$	$f(x)$的近似值
$\frac{\partial\varphi}{\partial \boldsymbol{n}}$	φ 沿方向 $\boldsymbol{n}$ 的方向导数
$\nabla\varphi/\mathbf{grad}\ \varphi$	φ 的梯度
$\Delta\varphi/\nabla^2\varphi$	拉普拉斯算子
$\Delta_n\varphi_{ij}$	离散拉普拉斯算子
$\boldsymbol{A}=(a_{ij})_n$	n 阶矩阵其元素为 a_{ij}
$(\boldsymbol{A}\mid\boldsymbol{b})$	$\boldsymbol{A}$ 的增广矩阵
$\boldsymbol{I}_n/\boldsymbol{I}$	n 阶单位矩阵
$\boldsymbol{A}^{\mathrm{H}}\overset{\mathrm{def}}{=\!=}\bar{\boldsymbol{A}}^{\mathrm{T}}$	矩阵 $\boldsymbol{A}$ 的共轭转置矩阵
$\boldsymbol{A}^{\mathrm{T}}$	矩阵 $\boldsymbol{A}$ 的转置矩阵
$\boldsymbol{A}^{-1}$	矩阵 $\boldsymbol{A}$ 的逆矩阵

$\mathrm{adj}\boldsymbol{A}/\boldsymbol{A}^{*}$	方阵 $\boldsymbol{A}$ 的伴随矩阵
$\{\boldsymbol{A}_k\}$	矩阵序列 $\boldsymbol{A}_1,\boldsymbol{A}_2,\cdots,\boldsymbol{A}_k,\cdots$
$\{\boldsymbol{x}_k\}$	向基序列 $\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_k,\cdots$
$\mathrm{diag}(d_1,d_2,\cdots,d_n)$	对角线元素为 $d_1,d_2,\cdots,d_n$ 的对角阵
$\det\boldsymbol{A}/\lvert\boldsymbol{A}\rvert$	方阵 $\boldsymbol{A}$ 的行列式
D_k	矩阵 $\boldsymbol{A}$ 的顺序主子式
$\sigma(\boldsymbol{A})$	矩阵 $\boldsymbol{A}$ 的谱
$\mathrm{tr}(\boldsymbol{A})$	矩阵 $\boldsymbol{A}$ 的迹
$\rho(\boldsymbol{A})$	矩阵 $\boldsymbol{A}$ 的谱半径
$\forall$	全称量词
$\exists$	存在量词
$\in$	属于
$\notin$	不属于
$\supset$	包含
$\subset$	包含于
$\cap$	交运算
$\cup$	并运算
$\backslash$	差运算
$\mathbb{C}$	复数集
$\mathbb{R}$	实数集
N	自然数集(包括零在内)
N^+	正自然数集
Z	整数集
Q	有理数集
$\mathbb{R}^n$	n 维实空间
$\mathbb{R}^{n\times m}$	$n\times m$ 维实空间
$\mathbb{C}^n$	n 维复空间
$\mathbb{C}^{n\times m}$	$n\times m$ 维复空间

$\text{span}\{\varphi_1, \varphi_2, \cdots, \varphi_N\}$	由 $\varphi_1, \varphi_2, \cdots, \varphi_N$ 生成的线性空间
Ω	解域
$\partial\Omega$	解域边界
Ω_k	离散解域
$\partial\Omega_k$	离散解域的边界
$C(\Omega)$	Ω 上全部连续函数的集合
$C^n(\Omega)$	Ω 上 n 阶连续可微函数的集合
sup	上确界
inf	下确界
$\delta(x)$	δ 函数
Δf	f 的向前差分
$\Delta^n f$	f 的 n 阶向前差分
∇f	f 的向后差分
$\nabla^n f$	f 的 n 阶向后差分
δf	f 的中心差分
I	不变算子：$If_k = f_k$
E	移位算子：$Ef_k = f_{k+1}$
$\Vert \cdot \Vert$	向量或矩阵的范数
$\text{cond}(\boldsymbol{A})$	矩阵 $\boldsymbol{A}$ 的条件数
$(\cdot, \cdot)$	函数内积或向量内积
$a \leftrightarrow b$	单元 a 和单元 b 交换
$r_i \leftrightarrow r_j$	矩阵的第 i 行与第 j 行交换
$\boldsymbol{a}[1:p]$	p 个元素的一维数组
$\boldsymbol{C}[1:n, 1:m]$	n 行 m 列矩形数组
$\boldsymbol{A}[*, i:j]$	矩阵 $\boldsymbol{A}$ 的第 i 列到第 j 列的所有元素
$\boldsymbol{A}[i:j, *]$	矩阵 $\boldsymbol{A}$ 的第 i 行到第 j 行的所有元素
O	$f(x)=O(g(x))$ 表示当 $x \to a$ 时，$f(x)/g(x)$ 有界

o	$f(x)=o(g(x))$表示当 $x\to a$ 时，$f(x)/g(x)\to 0$
mod	同余，$i=k(\mathrm{mod}n)$即 k 除以 n 余 i
$\dim M$	空间 M 的维数
H_n	n 阶多项式的集合
$\{x\mid P(x)\}$	使性质 $P(x)$成立的全体 x 组成的集合

目　　录

1　数值计算概论

1.1　数值分析的对象与特点

1.1.1　研究对象

数值分析即计算数学，是数学的一个分支，它的研究对象是利用计算机求解各种数学问题的数值方法及有关理论. 内容包括函数的数值逼近（代数插值与最佳逼近），数值积分与数值微分，非线性（代数的与超越的）方程（组）的数值解法，数值线代数（线性代数方程组的解法与矩阵特征值问题的计算），常微分方程的数值解法及偏微分方程的数值解法等.

数值分析也称计算方法，它所以成为数学中的独立分支，一方面是数学本身的发展为之提供了可能，另一方面是 20 世纪 40 年代电子计算机的问世使之成为必要. 在数学发展中，理论和计算是紧密联系的，逐渐积累了越来越多的数值方法. 因此，计算方法中不少方法是传统的（一些方法的命名就表明了这一点），它们在 19 世纪或 18 世纪，甚至更早一些时候就已建立. 现代计算机的出现为大规模的数值计算创造了条件，集中而系统地研究适用于计算机的数值方法立即变得十分迫切和必要. 数值分析并不仅仅是一些数值方法的简单积累，而且揭示包含在多种多样的数值方法之间的相同的结构和统一的原理. 它在大量的数值计算实践和理论分析工作的基础上迅速发展着，原有的方法有的使用至今，有的逐步被淘汰，而新的方法和新的理论不断地产生.

1.1.2 主要特点

数值分析有以下三个主要特点：

第一，面向计算机，提供实际可行的常用算法. 具体地讲，由于计算机能够进行加、减、乘、除四则运算，故需要把每个求解的数学问题用四则运算的有限形式的公式表达出来. 这种公式只能是多项式或有理分式的形式，通常称为算法，它是计算机能够直接处理的.

第二，能够任意逼近，有可靠的理论分析. 计算方法有两类算法，一类是精确的，另一类是近似的. 所谓精确算法是指在没有运算的舍入误差的假设下，能在确定的运算次数内获得数学问题的精确解. 近似算法本身有方法误差，从而在任何有限的运算次数内只能获得数学问题的近似解. 大多算法是近似算法. 由于计算机字长有限，每次运算都有舍入误差，因此无论精确算法还是近似算法都只能获得数学问题的近似解. 计算机需要人们用种种数学理论和方法来建立各个算法. 对近似算法要保证收敛性，即近似解能逼近精确解到任意的程度. 对每个算法要保证数值稳定性，这是指舍入误差对解的准确性影响不大.

第三，省时间省资源，有良好的计算复杂性. 一个算法的计算复杂性是指该算法包含的运算次数和所需的存储量. 近似算法的运算次数取决于算法的收敛速度. 求解一个数学问题，是否选用或建立计算复杂性好的算法很重要，有时会影响到在现有的计算机上能否真正实现.

从上述特点可以看出，数值分析的任务是提供在计算机上实际可行的、理论可靠的、计算复杂性好的各种算法.

1.1.3 数值问题与数值方法

数值问题是指输入数据(即问题中自变量与原始数据)与输出数据之间函数关系的一个确定描述，输入输出数据可用有限维向

量表示．根据这种定义，“数学问题”不一定是“数值问题”，它往往要用数值逼近方法才能转化为数值问题．函数的插值与逼近就是“数值分析”中最基本的问题之一，目的是提供各种简易的途径，将函数计算转化为数值问题，才能在计算机上处理．对于一个给定的数值问题有许多不同的算法，称为数值方法，它们都给出问题的近似答案，但所需计算量和得到的精度可能相差很大，必须选择面向计算机计算复杂性好并有可靠理论分析的数值方法．多数数学问题按建立数值方法的基本线索大体上可以归为两大类．

一类数学问题包含非有理函数或未知函数，如积分与微分计算、微分方程求解等．这类问题建立数值方法的基本线索是：首先利用函数的数值逼近或离散化将原问题化为数值问题，然后去计算或求解数值问题以得到原问题的近似值或近似解．例如，对利用各种方式得到的函数的逼近多项式进行求积分或求导数，可以引出各种形式的数值积分或数值微分公式；用一组离散点上待定值代替连续自变量的未知函数，通过各种逼近途径（包括插值、数值积分、数值微分以及泰勒（Taylor）展开等），将微分方程离散化，构成微分方程的各种数值解法．

另一类数学问题主要是代数问题，包括线性代数方程组的求解和矩阵特征值问题的计算，线性最小二乘法等不包含非有理函数，因其本身就是数值问题，可以直接去建立数值方法．非线性（代数的或超越的）方程的求解也可归于此类，因为它们仅仅在求函数值时可能要借助于数值逼近．这一类问题的数值方法大致可分为直接法、迭代法和变换约化法三种．直接法是一种精确算法，一般仅用于解线性代数方程组．迭代法的基本线索是针对确定类型的问题，寻求某种固定形式的递推公式，使得由公式产生的序列收敛于问题的解．迭代法在计算方法中占有重要的地位，某些数值问题如非线性方程等主要应用迭代法求解．目前，矩阵特征值问题的大多数重要的数值方法均利用相似变换将矩阵（近似地）约化为特殊

形式的矩阵，从而使特征值和特征向量可以较方便地近似求得，这类方法是变换迭代法，可称为变换约化法.

1.2 误差与有效数字

1.2.1 误差的来源与分类

应用数学解决实际问题时首先需要建立数学模型.一般地讲，数学模型是指那些利用数学语言模拟现实而建立起来的有关量的描述.数学模型通常总是近似的，其误差称为**模型误差**(model error).

数学模型中常包含某些参量，如质量、温度、电压等，此类量通过观测确定，产生的误差，称为**观测误差**(observational error).

例 1.2.1 假设 $L(t)$ 是金属棒在温度为 t 时的长度，其数学模型为

$$l(t)=1+\alpha t+\beta t^2,$$

其中 α,β 为参数.有如下估计：

$$\alpha=0.001253\pm10^{-6},\quad \beta=0.000068\pm10^{-6},$$

则 $L(t)-l(t)$ 是模型误差，10^{-6} 是 α 与 β 的观测误差.

数学模型常常不能获得精确解，必须用数值方法求近似解，其误差称为**截断误差**(truncation error)或**方法误差**.

例 1.2.2 实际计算时，函数 f 用泰勒多项式 P_n 近似代替：

$$P_n(x)=f(0)+\frac{f'(0)}{1!}x+\frac{f''(0)}{2!}x^2+\cdots+\frac{f^{(n)}(0)}{n!}x^n,$$

则数值方法的截断误差

$$R(x)=f(x)-P_n(x)=\frac{f^{(n+1)}(\xi)}{(n+1)!}x^{n+1},$$

其中，ξ 在 0 与 x 之间.

有了求解数学问题的算法后，由于计算机的字长有限，原始数

据在计算机上表示会产生误差，每一次运算又可能产生新的误差，这种误差称为**舍入误差**(tounding error)或**计算误差**.

例 1.2.3 实际计算时，用 3.14159 近似代替 π，则舍入误差

$$R=\pi-3.14159=0.0000026\cdots$$

数值分析中，仅讨论截断误差和舍入误差.

1.2.2 误差概念

定义 1.2.4 设 x 为准确值，a 为近似值，记

$$\Delta a = x - a, \quad \Delta_r a = \frac{\Delta a}{x} = \frac{x-a}{x},$$

称 Δa 为近似值 a 的**绝对误差**(absolute error)或**误差**，$\Delta_r a$ 为 a 的**相对误差**(relative error).

因为不能求得准确值 x 才去求近似值 a，所以，一般不能算出误差 Δa. 根据测量工具的精度或计算情况的分析，常常能够估计 $|\Delta a|$ 的较好上界. 用 δa 表示这种上界，它是非负数，称为**绝对误差界**. 于是

$$|\Delta a| = |x-a| \leqslant \delta a. \tag{1.2.1}$$

这样，就可知道准确值 x 所在的范围：$a-\delta a\leqslant x\leqslant a+\delta a$. 所以，只能说估计误差，而不说计算误差. 在实用上，常用下述写法来刻画 a 的精度：$x=a\pm\delta a$.

由于 x 未知，实际上总是将

$$\Delta_r^* a = \frac{\Delta a}{a} = \frac{x-a}{a} \tag{1.2.2}$$

作为 a 的相对误差. 记

$$\delta_r a = \frac{\delta a}{|a|}, \tag{1.2.3}$$

知道绝对误差界 δa 后便确定了 $\delta_r a$，显然，$\delta_r a$ 是 $|\Delta_r^* a|$ 的上界，称为 a 的**相对误差界**.

1.2.3 有效数字

在数值计算时,必须保证各个数据的每一位是可靠的,这是一切有意义的计算的前提.为此,需要建立有效数字的概念.

当 x 是准确值有很多位数时,常常需要按字长限制取 x 的位数来确定近似值 a,这个 a 按四舍五入规则来选取,能保证绝对误差最小.例如

$$x=\pi=3.14159265\cdots$$

取 3 位,$a=3.14$,在所有 3 位数中 3.14 与 π 的误差最小;取 5 位,$a=3.1416$,在所有 5 位数中 3.1416 与 π 误差最小.它们的误差均不超过末位的半个单位,即

$$|\pi-3.14|\leqslant\frac{1}{2}\times10^{-2},\quad |\pi-3.1416|\leqslant\frac{1}{2}\times10^{-4}.$$

定义 1.2.5 设数 x 的近似值

$$a=\pm 10^k\times0.a_1a_2\cdots a_n,$$

其中,每个 $a_i(i=1,2,\cdots,n)$ 是 0 到 9 中的一个数字,$a_1\neq0$.如果 a 的误差不超过末位的半个单位,即

$$|x-a|\leqslant\frac{1}{2}\times10^{k-n},$$

则称用 a 近似 x 时具有 $\boldsymbol{n}$ **位有效数字**(significant digits).

实际上,有效数字的概念无非是说:按四舍五入规则得到的近似值,其每一位数都是有效的.

显然,有效数字位数与小数点的位置无关.有效位数越多,绝对误差和相对误差就都越小.

有了有效数字概念后,以下写法是有区别的:

$$34.01,\quad 34.0100,$$

前者是 4 位有效数字,后者则表示 6 位有效数字.

1.3 误差估计与误差分析

1.3.1 算术运算的误差界

定理 1.3.1 设 x 与 y 是精确值，a 与 b 是相应的近似值，绝对误差界分别为 δa 与 δb，则

$$\delta(a \pm b) = \delta a + \delta b,$$

$$\delta(ab) \approx |a| \delta b + |b| \delta a,$$

$$\delta\left(\frac{a}{b}\right) \approx \frac{|a| \delta b + |b| \delta a}{|b|^2} \quad (b \neq 0).$$

例 1.3.2 $\alpha = 1.21 \times 3.65 + 9.81$，其中每个数据的绝对误差界为 0.005. α 的绝对误差界

$$\begin{aligned} \delta(\alpha) &= \delta(1.21 \times 3.65) + \delta(9.81) \\ &\approx 1.21 \times 0.005 + 3.65 \times 0.005 + 0.005 \\ &= 0.0293 \leqslant 0.03. \end{aligned}$$

定理 1.3.3 设 a 与 b 是近似值，相对误差界分别为 $\delta_r a$ 与 $\delta_r b$，则

$$\delta_r(a + b) = \max\{\delta_r a, \delta_r b\} \quad (a 与 b 同号),$$

$$\delta_r(a - b) = \frac{|a| \delta_r a + |b| \delta_r b}{|a - b|} \quad (a 与 b 同号),$$

$$\delta_r(ab) \approx \delta_r a + \delta_r b,$$

$$\delta_r\left(\frac{a}{b}\right) \approx \delta_r a + \delta_r b \quad (b \neq 0).$$

例 1.3.4 例 1.3.2 中 α 的相对误差界

$$\begin{aligned} \delta_r(\alpha) &= \max\{\delta_r(1.21 \times 3.65), \delta_r(9.81)\} \\ &\approx \max\{\delta_r(1.21) + \delta_r(3.65), \delta_r(9.81)\} \\ &= \max\left\{\frac{\delta(1.21)}{1.21} + \frac{\delta(3.65)}{3.65}, \frac{\delta(9.81)}{9.81}\right\} \end{aligned}$$

$$= \max\left\{\frac{0.005}{1.21}+\frac{0.005}{3.65},\frac{0.005}{9.81}\right\}$$
$$\approx \max\{0.0055, 0.0005\}$$
$$= 0.0055.$$

1.3.2 函数求值的误差估计

在计算一元或多元函数的值时，由于自变量数据不精确会产生误差.

设 f 是一元函数，要计算在点 x 的函数值，但仅知 x 的近似值为 a，以 $f(a)$ 近似 $f(x)$，$f(a)$ 的绝对误差界 $\delta(f(a))$ 可用泰勒公式估计，假定 f 在包含 x 与 a 的一个开区间上存在足够阶的导数，则有

$$\Delta(f(a)) = f(x) - f(a) = f'(a)(x-a) + \frac{f''(\xi)}{2!}(x-a)^2,$$

其中 ξ 在 x 与 a 之间，取绝对值，得

$$|\Delta(f(a))| = |f(x) - f(a)|$$
$$\leqslant |f'(a)|\delta a + \frac{|f''(\xi)|}{2}(\delta a)^2,$$

假定 $|f''(a)|$ 与 $|f'(a)|$ 相比不太大，则可以忽略高阶项，得

$$\delta(f(a)) \approx |f'(a)|\delta a. \tag{1.3.1}$$

但是，如果 $|f'(a)|$ 是零或值很小，则要考虑后面的项. 特别地，若

$$f'(a) = f''(a) = \cdots = f^{(k-1)}(a) = 0, \quad f^{(k)}(a) \neq 0,$$

且 $|f^{(k+1)}(\xi)|$（ξ 在 x 与 a 之间任意变动）不很大，则

$$\delta(f(a)) \approx \frac{|f^{(k)}(a)|}{k!}(\delta a)^k. \tag{1.3.2}$$

例 1.3.5 设 $f(x)=\sin x$，$\alpha=45^\circ$，$\beta=90^\circ$，$\delta\alpha=0.1^\circ$，$\delta\beta=0.2^\circ$. 由于

$$f'(\alpha) = \cos\alpha = \frac{\sqrt{2}}{2}, \quad f'(\beta) = 0, \quad f''(\beta) = -\sin\beta = -1,$$

所以，在把 $\delta\alpha$ 与 $\delta\beta$ 化为弧度后有

$$\delta(\sin\alpha) \approx |\cos\alpha| \delta\alpha = \frac{\sqrt{2}}{2} \times 0.1 \times \frac{\pi}{180} \approx 1.2 \times 10^{-3},$$

$$\delta(\sin\beta) \approx \frac{|\sin\beta|}{2!}(\delta\beta)^2 = \frac{1}{2} \times \left(0.2 \times \frac{\pi}{180}\right)^2 \approx 6.1 \times 10^{-6}.$$

多元函数值的误差界可用多元函数的泰勒公式得到：

$$\delta\Big(f(a_1, a_2, \cdots, a_n)\Big) \approx \sum_{i=1}^{n} \left| \frac{\partial f(a_1, a_2, \cdots, a_n)}{\partial x_i} \right| \delta a_i. \quad (1.3.3)$$

如果一阶偏导数绝对值都很小或等于零，则要利用高阶项，其一般描述与一元函数类似.

例 1.3.6 设 $f(x, y) = xy$，a 和 b 分别是 x 和 y 的近似值，则

$$\begin{aligned}\delta(f(a,b)) &\approx |f_x(a,b)| \delta a + |f_y(a,b)| \delta b \\ &= |b| \delta a + |a| \delta b.\end{aligned}$$

1.3.3 误差分析方法

误差分析是指数值计算中舍入误差的分析，它是一个重要而复杂的问题，前面讨论的不精确数据运算结果的误差界只能适用于运算次数少的简单情形. 对于工程与科学计算，由于运算次数往往以千万计，且原始数据有误差，每步运算会产生新的舍入误差并传播前面各数据的误差，每步误差有正有负，都按其上界估计是不合理的，所以按步分析是办不到的. 如何解决这个问题目前尚无有效理论. 已提出的误差分析方法有以下几种：

向前误差分析法与向后误差分析法是分析算法舍入误差积累（不涉及截断误差）的两种不同方法. 假设所讨论的算法由若干公式表达，某个新的量 x 由已知量（前面已算出的量或原始数据）a_1，a_2，…，a_n 的某个公式定义，写成

$$x = g(a_1, a_2, \cdots, a_n),$$

x 从 $a_1,a_2,\cdots,a_n$ 经过基本的算术运算得出. 因为计算中产生舍入误差,实际计算值记作 x_{fl}(用 fl 表示是由计算机浮点计算得到的),它与精确值 x 不同. **向前误差分析**(forward error analysis)是对每一步计算找出舍入误差界,随计算过程逐步向前分析,直至估计出最后结果的舍入误差 $|x-x_{fl}|$ 的界. **向后误差分析**(backward error analysis)则是把舍入误差与导出 x_{fl} 的已知量 $a_1,a_2,\cdots,a_n$ 的某种摄动(误差)等价起来,即对每个 a_i 引进某个摄动量 ϵ_i,使得精确地成立

$$x_{fl}=g(a_1+\epsilon_1,a_2+\epsilon_2,\cdots,a_n+\epsilon_n),$$

并推出这些 ϵ_i 的界(并非要得出 ϵ_i 的具体值,ϵ_i 不是惟一的),然后利用摄动理论估计最后的舍入误差界. 向后误差分析法是一种先验估计法,特别在数值线代数(矩阵运算)的误差研究中有比较系统的应用,取得了较大的进展. 相比之下,向前误差分析法只能应用于十分简单的情形.

区间分析法是出现不久的一种研究误差的方法,它主要利用区间分析这一数学新分支中的区间运算理论. 设 x,y 是准确值,α,β 是相应的近似值,且已知绝对误差界 $\delta\alpha,\delta\beta$,则能确定 x,y 的所在区间

$$\alpha-\delta\alpha\leqslant x\leqslant\alpha+\delta\alpha,\quad \beta-\delta\beta\leqslant x\leqslant\beta+\delta\beta,$$

或者写成

$$x\in[\alpha-\delta\alpha,\alpha+\delta\alpha],\quad y\in[\beta-\delta\beta,\beta+\delta\beta],$$

这样,利用 x,y 的所在区间,按照区间分析中的区间运算,能够得出 x 与 y 之间各种运算的精确结果的所在区间,并由这个区间给出实际运算结果的误差估计,实际运算自然是在 α 与 β 之间进行的. 这就是区间分析法的基本思想.

前面提供的关于不精确数据运算结果的误差界,及舍入误差分析引出的误差界,通常远远大于实际的误差. 实际上误差分布有随机性,不会经常地达到上界. 因此,利用概率和统计方法,将数据

和运算中的误差视为适合某种分布的随机变量,然后确定计算结果的误差分布,并用它代替绝对误差界,常常可使误差估计更接近实际,这种误差分布称为**概率界**,这种分析方法就是**概率分析法**.

这些方法在实际误差估计中都不太可行,因此在数值计算中通常更着重误差的定性分析,即研究数值算法的数值稳定性和数值问题本身是否病态,以及数值运算中避免误差危害的原则,而不具体估计舍入误差界.

1.4 误差的定性分析与运算原则

1.4.1 算法的数值稳定性

定义 1.4.1 一个算法如果输入数据有误差,而计算过程中舍入误差不增长,则称此算法是**数值稳定的**,否则算法称为**不稳定的**.

例 1.4.2 计算积分 $y_n = \int_0^1 \frac{x^n}{x+5}\mathrm{d}x \ (n=0,1,\cdots,8)$ 可利用递推公式 $y_n + 5y_{n-1} = \frac{1}{n}$,其中 $y_0 = \int_0^1 \frac{1}{x+5}\mathrm{d}x = \ln\frac{6}{5}$.

解 利用递推公式

$$y_n = \frac{1}{n} - 5y_{n-1} \quad (n = 0,1,\cdots,8), \tag{1.4.1}$$

计算 y_n,若算到小数点后 3 位,由于初始值 $y_0 = \ln\frac{6}{5} \approx 0.182 = \bar{y}_0$,用 $\bar{y}_0$ 近似 y_0,误差为 $\varepsilon_0 = y_0 - \bar{y}_0$,由式(1.4.1)得

$$\bar{y}_n = \frac{1}{n} - 5\bar{y}_{n-1} \quad (n = 1,2,\cdots,8). \tag{1.4.2}$$

计算结果见表 1.1,表中 y_n 是误差小于 $\frac{1}{2}\times10^{-6}$ 的积分近似值,从表中看到 $\bar{y}_n$ 的误差很大,在 $n=4$ 就已完全失真,因为 $y_n>0$,

而 $\bar{y}_4<0$，显然不对. 这里计算公式是正确的，只是因为初始近似 $\bar{y}_0$ 有误差 ε_0，而 $|\varepsilon_0|=|y_0-\bar{y}_0|<\frac{1}{2}\times 10^{-3}$，令 $\varepsilon_n=y_n-\bar{y}_n$，由式(1.4.1)减式(1.4.2)得

$$\varepsilon_n=-5\varepsilon_{n-1}=\cdots=(-5)^n\varepsilon_0, \quad |\varepsilon_n|\geqslant 5^n|\varepsilon_0|.$$

它表明误差增长很快，故算法(1.4.1)是不稳定的.

表 1.1 积分 y_n 与近似值 $\bar{y}_n$ 与 $\bar{y}_n$ 比较

a	y_n	$\bar{y}_n$	$\bar{y}_n$
0	0.182322	0.182	0.182
1	0.088392	0.090	0.088
2	0.058039	0.050	0.058
3	0.043139	0.083	0.043
4	0.034306	−0.165	0.034
5	0.028468	1.025	0.028
6	0.024325	−4.958	0.025
7	0.021231	24.933	0.021
8	0.018846	−124.540	0.019

若将公式(1.4.1)改写为

$$y_{n-1}=\frac{1}{5}\left(\frac{1}{n}-y_n\right) \quad (n=8,7,6,\cdots,1), \qquad (1.4.3)$$

取 $\bar{y}_8=0.019$，$\bar{\varepsilon}_8=|y_8-\bar{y}_8|<\frac{1}{2}\times 10^{-3}$，由式(1.4.3)计算出 $\bar{y}_{n-1}=\frac{1}{5}\left(\frac{1}{n}-\bar{y}_n\right)(n=8,7,\cdots,1)$，结果见表 1.1，此时 $\bar{\varepsilon}_{n-1}=-\frac{1}{5}\bar{\varepsilon}_n$，$\bar{\varepsilon}_{8-k}=(-1)^k\left(\frac{1}{5}\right)^k\bar{\varepsilon}_8$，误差是逐步递减的. 故用算法(1.4.3)计算是数值稳定的. 在数值计算中数值不稳定的算法是不能使用的.

1.4.2 病态问题与条件数

数值稳定性是针对算法而言,对数值问题本身如果输入数据有微小变动引起输出数据(即问题的解)很大扰动(误差),就是病态问题,它是数学问题本身决定的,与数值问题算法无关.

例 1.4.3 求方程组$\begin{cases}x+\alpha y=1\\ \alpha x+y=0\end{cases}$的解.

解 当$\alpha=1$时,系数矩阵奇异,解不存在.当$\alpha\neq 1$时解为$x=\dfrac{1}{1-\alpha^2}$, $y=\dfrac{-\alpha}{1-\alpha^2}$.若输入数据$\alpha$有误差,且$\alpha\approx 1$,则解的误差很大.例如$\alpha=0.99$时的解$x=50.25$,若$\alpha$有误差0.001,即$\tilde{\alpha}=0.991$,则解$\tilde{x}\approx 55.81$,解误差$\tilde{x}-x\approx 5.56$.所以在此例中若$\alpha\approx 1$则问题是病态的.它与选用何种求解方法无关.

如何判断问题是否病态呢?通常可用条件数衡量(关于解线性方程组的条件数将在第6章介绍).这里只以计算函数$f(x)$的值为例给出条件数定义:

$$\begin{aligned}L_p &= \mathrm{cond}(f(x))\\ &= \lim_{\Delta x\to 0}\left|\frac{f(x+\Delta x)-f(x)}{f(x)}\Big/\frac{\Delta x}{x}\right|\\ &= \left|\frac{xf'(x)}{f(x)}\right|,\end{aligned}\tag{1.4.4}$$

它是计算函数$f(x)$的相对误差$\left|\dfrac{f(x+\Delta x)-f(x)}{f(x)}\right|$与$x$相对误差之比的极限.根据定义相对误差越大条件数$C_p$也越大,病态越严重,通常$C_p\gg 1$就认为问题病态.在此例中$x=\dfrac{1}{1-\alpha^2}$是$\alpha$的函数,利用式(1.4.4)的结果,由于$x'(\alpha)=-\dfrac{2\alpha}{(1-\alpha^2)^2}$,故$C_p=\left|\dfrac{\alpha x'(\alpha)}{x(\alpha)}\right|=\dfrac{2\alpha^2}{1-\alpha^2}$.当$\alpha=0.99$时$C_p\approx 100$,故问题是病态的.

1.4.3 数值运算的简单原则

为了减少数值运算中的舍入误差，应注意避免误差危害，通常可采用以下简单原则.

(1) 注意运算次序

在计算机中，由于字长的限制，$(a+b)+c$ 可能不等于 $a+(b+c)$，因为两者的舍入误差可能不同. 例如，以限制取两位十进制数为例，$(10^1\times0.19+10^{-1}\times0.43)+10^{-1}\times0.47$ 的两位近似值为 $10^1\times0.19$，舍入误差为 $10^{-1}\times0.90$；而 $10^1\times0.19+(10^{-1}\times0.43+10^{-1}\times0.47)$的两位近似值为 $10^1\times0.20$，舍入误差为 $10^{-1}\times0.10$. 由此例可以看出一个简单原则：在多个数求和时，如果被加数的绝对值之间差异较大，且包含许多绝对值较小的数，则应按绝对值从小到大的次序相加.

例 1.4.4 在 4 位十进制的限制下，计算

$$A=1000+\delta_1+\delta_2+\cdots+\delta_{1000},$$

其中 $0.1\leqslant\delta_i\leqslant0.4$，$i=1,2,\cdots,1000$.

解 如果自左到右相加，则每个 δ_i 被 1000"吃掉"，A 的 4 位近似值为 1000；如果先把所有 δ_i 加起来再加 1000，则

$$\begin{aligned}1100=&1000+0.1\times1000\\ &\leqslant A\leqslant1000+0.4\times1000\\ &=1400.\end{aligned}$$

(2) 尽量避免相近数相减

相近数相减会使有效数字大量丢失，如有可能应尽量避免. 主要是在构造具体算法时要周密地考虑，防止产生这样的情况.

例 1.4.5 二次方程 $ax^2+bx+c=0$ 根的公式的通常形式为

$$x_1=\frac{-b+\sqrt{b^2-4ac}}{2a},\quad x_2=\frac{-b-\sqrt{b^2-4ac}}{2a},$$

当遇到 $b^2\gg4|ac|$ 的情形时，有 $|b|\approx\sqrt{b^2-4ac}$，用上述公式求出

的两个根中，总有一个因相近数相减而严重不可靠. 为了求得可靠的结果，鉴于 $x_1x_2=c/a$，在计算机上采用如下算法：

$$x_1=\frac{-b-\operatorname{sgn}(b)\sqrt{b^2-4ac}}{2a},\quad x_2=\frac{c}{ax_1},$$

这里的 x_1 与上面的 x_1，当 $b<0$ 时是相同的. 这个公式避免了相近数相减的可能性.

例 1.4.6 计算 $A=1-\cos 2^\circ$(用四位数学用表).

解 由于 $\cos 2^\circ=0.9994$(四位有效数字)，直接计算有

$$A=1-\cos 2^\circ=1-0.9994=0.0006,$$

只有一位有效数字，若利用 $1-\cos x=2\sin^2\frac{x}{2}$，则 $A=1-\cos 2^\circ=2(\sin 1^\circ)^2=2(0.0175)^2=6.13\times10^{-4}$，具有三位有效数字. 此例表明通过改变计算公式可避免或减少有效数字的损失.

(3) 避免被除数绝对值远远大于除数绝对值的除法

绝对值大的数被绝对值小的数除，舍入误差界要比相反的情形大，有时会给计算结果带来严重的影响.

例 1.4.7 求解二元线性方程组

$$\begin{cases}0.0001x_1+x_2=1,\\x_1+x_2=2.\end{cases}$$

解 此方程组的精确解为

$$x_1=\frac{10000}{9999},\quad x_2=\frac{9998}{9999}.$$

下面在 3 位十进制的限制下用消去法求解，上述方程组应改写成

$$\begin{cases}10^{-3}\times0.100x_1+10^1\times0.100x_2=10^1\times0.100,\\10^1\times0.100x_1+10^1\times0.100x_2=10^1\times0.200.\end{cases}$$

若利用第一个方程消去第二个方程中含 x_1 的项，将第二个方程减去第一个方程的 $(10^{-3}\times0.100)^{-1}$ 倍，则出现大数被小数除的情形，得到

$$\begin{cases}10^{-3}\times 0.100x_1+10^1\times 0.100x_2=10^1\times 0.100,\\ -10^5\times 0.100x_2=-10^5\times 0.100.\end{cases}$$

由此解出

$$x_1=0(\text{失去近似意义}),\quad x_2=10^1\times 0.100.$$

若反过来用第二个方程消去第一个方程中含 x_1 的项，则避免出现大数被小数除的情形，得到

$$\begin{cases}10^1\times 0.100x_2=10^1\times 0.100,\\ 10^1\times 0.100x_1+10^1\times 0.100x_2=10^1\times 0.200.\end{cases}$$

由此得出相当好的近似解

$$x_1=x_2=10^1\times 0.100.$$

(4) 简化计算步骤

每次算术运算都可能产生舍入误差，因此，如能通过算法的改进减少运算次数，特别是减少乘除法的运算次数，则舍入误差的积累一般可能下降，还能节省计算机的执行时间.

算法是通过在具体执行上没有任何随意性的表达式来描述的计算过程，因此不是任何一个表达式都确定一个算法.

例 1.4.8 求多项式

$$P_n(x)=a_nx^n+a_{n-1}x^{n-1}+\cdots+a_1x+a_0$$

的值.

解 多项式是一个表达式，但由它导出结果的过程有很大的随意性，它不是一个算法而是一个计算问题. 若直接计算 a_kx^k 再逐项相加，则确定一种算法：

$$\begin{cases}A_0=a_0,\\ A_k=a_kx^k\quad(k=1,2,\cdots,n),\\ P_n(x)=A_0+A_1+\cdots+A_n.\end{cases}$$

这个算法有明显的缺点，比如 x 的幂的重复计算，它共需做

$$1+2+\cdots+n=\frac{n(n+1)}{2}$$

次乘法和 n 次加法.

若用著名的**秦九韶算法**或称 **Horner 算法**

$$\begin{cases} S_n = a_n \\ S_k = xS_{k+1} + a_k \quad (k = n-1, \cdots, 2, 1, 0) \\ P_n(x) = S_0 \end{cases} \tag{1.4.5}$$

求 $P_n(x)$ 的值只需做 n 次乘法和 n 次加法.

例 1.4.9 $\frac{2}{3}=\frac{1}{3}\times 2$,在 4 位十进制的限制下,$\frac{2}{3}$ 的近似值为 $10^0\times 0.6667$,舍入误差为 $10^{-4}\times 0.44\cdots$; $\frac{1}{3}\times 2$ 比 $\frac{2}{3}$ 多一次运算,近似值为 $10^0\times 0.6666$,舍入误差为 $10^{-4}\times 0.66\cdots$

1.5 并行算法及其基本概念

1.5.1 并行算法及其分类

传统的数值计算方法是只有一个进程的算法,称为**串行算法**.它适合串行计算机,其特点是每一时刻按一条指令加工处理一个或一对数据,这类计算机称为**单指令流单数据流系统**,简称 SISD (single instruction stream single data stream)型,并行计算机与此不同,它每一时刻可按多条指令处理多个数据,必须包含两个以上处理机.面向并行计算机具有两个以上进程的算法称为**并行算法**(parallel algorithms).

例 1.5.1 求 N 个数 $a_1, a_2, \cdots, a_N$ 的和

$$s = \sum_{i=1}^{N} a_i.$$

解 一个进程的串行算法是累加算法

$$s_1 = a_1, \quad s_k = s_{k-1} + a_k, \quad k = 2, \cdots, N,$$

按指令流是单个还是多个，并行算法可分为两大类：SIMD算法与MIMD算法.

SIMD算法的基本特征是：

(1) k个进程由一个指令流控制，一个并行步仅由一条指令控制，故每个并行步所含的操作必须完全相同.

(2) 允许并行步中操作的个数在2至k之间有任意性，且有含k个操作的并行步.

(3) 能在一个具有k个处理机的SIMD系统中实现.

例 1.5.2 给出的扇入加法就是SIMD算法.

MIMD算法的基本特征是：

(1) k个进程由k个指令流控制，故每个并行步所含各操作可两两相异，而且存在包含不同操作的并行步.

(2) 同SIMD算法的特征(2).

(3) 能在一个具有k个处理机的MIMD系统中实现.

按照进程之间是否需要同步也可将并行算法分为同步算法与异步算法.

同步算法 是指在k个进程的算法中有些进程的若干算法必须在另一些进程的某些算法之后执行，为此有些进程可能出现在计算之前或计算之间的等待阶段. 同步算法可在一个具有k个处理机的SIMD系统或MIMD系统中实现.

异步算法 k个进程间有信息联系但不需同步，它只能在一个具有k个处理机的MIMD系统中实现. 因此异步算法一定是MIMD算法. 使用这种算法时各次执行的实际过程可能互不重复. 同步算法既可以是SIMD算法也可以是MIMD算法，而SIMD算法一定是同步算法，进程间的同步通过单指令流的控制来保证. 这样，并行算法可分为三类，SIMD算法，同步MIMD算法和异步算法.

例 1.5.3 对$f(x)=0$求根的牛顿(Newton)法

$s_N=s$,即为所求.在累加过程中所给和数逐次减1,显然它不适合于并行计算.图1.1给出 $N=8$ 时的一种并行算法,称为**扇入加法**.它可组成4个进程 P_1,P_2,P_3,P_4.

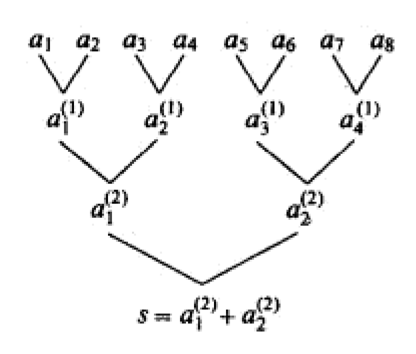


图1.1 扇入加法示意图

P_1	P_2	P_3	P_4
$a_1^{(1)}=a_1+a_2$	$a_2^{(1)}=a_3+a_4$	$a_3^{(1)}=a_5+a_6$	$a_4^{(1)}=a_7+a_8$
$a_1^{(2)}=a_1^{(1)}+a_2^{(1)}$		$a_2^{(2)}=a_3^{(1)}+a_4^{(1)}$	
$s=a_1^{(2)}+a_2^{(2)}$			

在有4个处理机的并行系统中用 $3=\log_2 8$ 级完成8个数求和,第1级并行做4个加法,第2级并行做2个加法,第3级做1个加法.

并行计算机有不同类型,用一个控制器控制多个处理机,每一时刻都执行同一指令对单个或一对数组进行同样加工,这类并行机称为**单指令流多数据流系统**.简称SIMD(single instruction stream multiple data stream)型.另一类并行机由多个控制器分别控制多个处理机.各处理机在自己的指令控制下处理自己的数据流,称为**多指令流多数据流系统**,简称MIMD(multiple instruction stream multiple data stream)型.

P_1	P_2
$f(t_3)\to t_1$	$f'(t_3)\to t_2$
$t_3-t_1/t_2\to t_3$,检验	
$f(t_3)\to t_1$	$f'(t_3)\to t_2$
$t_3-t_1/t_2\to t_3$,检验	
$f(t_3)\to t_1$	$f'(t_3)\to t_2$
$t_3-t_1/t_2\to t_3$,检验	

图 1.3　异步算法示意图

其一般关系如下：

$$x_{k+1}=x_k-f(x_k)/f'(x_j)\quad(k=0,1,\cdots,j<k).\tag{1.5.2}$$

当 $k\to\infty$ 时 $j\to\infty$，这与传统牛顿法(1.5.1)不同. 它是一种混乱迭代，其收敛性要另外证明. 由于计算 $f'(x)$ 比 $f(x)$ 更费时间，所以，只要 $f'(x_j)$ 计算出新值就用于迭代，由于异步算法不用等待，故更节省时间.

1.5.2　并行算法基本概念及设计原则

评价和分析并行算法，主要应关注它的计算复杂性，即算法的运行时间(时间复杂性)与所提供的处理机个数(空间复杂性). 并行处理的基本思想是增加处理机个数来换取运算时间的节省. 为评价并行算法需引进一些基本概念.

定义 1.5.4　一个算法的**并行度**是指算法中能用一个运算步(并行)完成的运算个数，假设算法总运算个数为 r，用 s 个运算步完成，则 $\frac{r}{s}$ 称为**平均并行度**.

如例 1.5.1 中 N 个数求和，若用串行算法需要 $N-1$ 个加法步，每步一个加法，并行度为 1，总运算个数为 $N-1$，平均并行度为 $\frac{N-1}{N-1}=1$；如果用扇入加法，对 $N=2^n$ 个数求和，有 $n=\log_2 N$

个加法步，并行性由高到低分布，逐步减半. 运算个数 $r=\frac{N}{2}+\frac{N}{2^2}+\cdots+1=2^n-1$，运算步数 $s=n=\log_2 N$，平均并行度为

$$\frac{r}{s}=\frac{N-1}{\log_2 N}=O\left(\frac{N}{\log_2 N}\right).$$

N 个数求和的 $N-1$ 个加法不能用一个并行步完成，即不能以 $N-1$ 为并行度，但用扇入加法可把并行度降为 $\frac{N-1}{\log_2 N}$.

并行度和平均并行度是算法内在并行性的一种度量，不依赖于并行系统中处理机的个数，当然并行机的个数会影响完成计算所需时间.

处理机的个数充分多时的最少运行时间称为算法的**时间界**，而算法的运行时间达到时间界时需要提供的处理机个数，称作**处理机个数界**. 上述 N 个数求和的时间界为 $T=\log_2 N$，而处理机个数界为 $\frac{N}{2}$.

考察 p 台处理机组成的并行系统，假定每个处理机的算术运算的操作时间相同，可引进加速比和效率的定义作为并行化的度量.

定义 1.5.5 设 T_1 为串行算法在单处理机上运行时间，T_p 为并行算法在 p 个处理机系统上的运行时间，则一个算法的加速比定义为

$$s_p=\frac{T_1}{T_p},$$

该算法的效率为

$$E_p=\frac{s_p}{p}.$$

加速比和效率是评价并行算法的重要指标，研究并行算法的目标是达到尽可能大的加速比，理想上 $s_p=p$，这时并行效率达到

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \quad (k = 0,1,\cdots). \qquad (1.5.1)$$

分别设计两个进程的同步及异步算法.

解 同步算法：可将每次迭代分成三个计算元，分别计算 $f(x_k) \to f_k, f'(x_k) \to f'_k, x_k - \frac{f_k}{f'_k} \to x_{k+1}$ 及检验精度. 将计算分为两个进程 P_1 及 P_2，假定计算 $f'(x_k)$ 的时间比计算 $f(x_k)$ 时间长，则两个进程或其中一个必出现等待继续计算所需数据的情况，图 1.2 给出两种同步 MIMD 算法示意图.

(a)

P_1	P_2
$f(x_0) \to f_0$	$f'(x_0) \to f'_0$
等待 f'_0	
$x_0 - f_0/f'_0 \to x_1$ 检验精度	等待 x_1
$f(x_1) \to f_1$	$f'(x_1) \to f'_1$
等待 f'_1	
$x_1 - f_1/f'_1 \to x_2$ 检验精度	等待 x_2

(b)

P_1	P_2
$f(x_0) \to f_0$	$f'(x_0) \to f'_0$
等待 x_1	$x_0 - f_0/f'_0 \to x_1$ 检验精度
$f(x_1) \to f_1$	$f'(x_1) \to f'_1$
等待 x_2	$x_1 - f_1/f'_1 \to x_2$

图 1.2　MIMD 算法示意图

异步算法，可引进 3 个公用变量 t_1, t_2, t_3 分别表示 $f(x)$，$f'(x)$ 及 x 在计算中的当前值，仍假定计算 $f'(x)$ 比计算 $f(x)$ 更费时间. 图 1.3 给出两个进程的异步算法，进程 P_1 更新 t_1 与 t_3，且检验根的近似值是否满足精度要求，进程 P_2 更新 t_2，假定变量初值 $t_1 = 0, t_2 = c \neq 0, t_3 = x_0$，前三个近似值为

$$x_1 = x_0 - f(x_0)/c,$$
$$x_2 = x_1 - f(x_1)/f'(x_0),$$
$$x_3 = x_2 - f(x_2)/f'(x_1).$$

$E_p=1$，但一般不可能达到理想的加速比 $s_p=p$，这是因为：

(1) 算法缺乏必要的并行度；

(2) 在并行系统上没有达到完全的负荷均衡；

(3) 通信、存储争用及同步时间延迟等.

设计并行算法使其加速比尽可能大，一个重要原则是"分而治之"，其基本思想是把问题依次划分为可以独立完成的较小问题，将规模逐次减半的二分技术就是并行算法设计的一种基本技术，例 1.5.1 给出的求和的扇入加法就是一种二分技术. 用二分技术可将很多在串行机上传统的串行算法改造成适合并行计算的算法，如矩阵计算及解方程组的直接方法. 在迭代法中有些本身具有直接的并行性，如解方程组的雅可比(Jacobi)迭代法. 还有些迭代法经过重新排序也可改造成适合并行的算法，异步算法的混乱迭代法是并行算法最富有特色的组成部分之一，混乱迭代不是传统意义的迭代法，在理论上必须做收敛性与收敛速度的分析.

根据并行算法特点设计具有新思想的新算法是近 20 年并行算法发展更值得重视的方向，它的出发点仍然是"分而治之"的原理，符合此原理的区域、算子、系统的分裂方法和技术是设计和实现并行处理的重要手段. 如解线性与非线性方程组的多分裂方法，解偏微分方程的区域分解法(即 DDM 方法)，都属于此类. 由于这些方法减小了求解方程组的规模，使计算时间大为减少，因此实际效率大为提高.

2 插值法

2.1 引言

2.1.1 插值的意义

函数插值的提出源于关于函数的两方面问题.一方面,在实际问题中出现的函数,常有赖于实验和观察,虽然可能在某个区间上有定义,却没有明确的表达式,而只能得到区间内一些离散点上的值,因此希望对这样的函数能用简单表达式近似地给出整体上的描述,并能与已知的离散点上的值相符.另一方面,函数有明确的表达式,但只要不是(分段)有理函数,便不易计算和使用,这样,也需要用简单的函数提供一个好的逼近.

在插值的研究工作中,对用于逼近的简单函数的类型有不同的选取.多项式或分段多项式最便于计算和使用,因而最引人注目.特别是计算机出现后,人们的注意力更集中在利用多项式的插值方面,这是因为计算公式相对地易于描述和进行程序设计,其误差分析也比较简单.插值还是建立适用于计算机的许多其他算法的一个基本工具.

2.1.2 插值问题的提法

设 f 是区间 $[a,b]$ 上的一个实函数,且已知离散数据

$$(x_i, y_i), y_i = f(x_i) \quad (i = 0,1,\cdots,n),$$

其中 $x_0, x_1, \cdots, x_n$ 是 $[a,b]$ 上的 $n+1$ 个相异的实数,即 $a \leqslant x_0 < x_1 < \cdots < x_n \leqslant b$.

插值问题最基本的提法是：寻求一个次数尽可能低的多项式 p，满足条件

$$p(x_i) = y_i \quad (i = 0,1,\cdots,n). \tag{2.1.1}$$

从几何上看，就是寻求一个最低次的多项式，其几何曲线通过给定的 $n+1$ 个点 $(x_i, y_i)(i=0,1,\cdots,n)$.

如果所说的多项式 p 存在，并用它近似函数 f（p 与 f 有 $n+1$ 个相同的值），则称 p 为 f 的**插值多项式**（interpolation polynomial），$x_0, x_1, \cdots, x_n$ 称为**插值节点**或**简称节点**（node），$[a,b]$称为**插值区间**，条件(2.1.1)称为**插值条件**，f 称为**被插函数**（interpolated function）.

插值问题有进一步的提法. 一种提法是寻求满足插值条件(2.1.1)的分段多项式；另一种提法是插值条件中增加在某些节点上的导数值的条件，寻求相应的多项式.

更一般的插值问题是寻求一个函数，形如

$$\varphi(x) = a_0\varphi_0(x) + a_1\varphi_1(x) + \cdots + a_n\varphi_n(x), \tag{2.1.2}$$

其中 $a_0, a_1, \cdots, a_n$ 为待定参量，$\varphi_0, \varphi_1, \cdots, \varphi_n$ 是$[a,b]$上 $n+1$ 个线性无关的连续函数，使 $\varphi(x)$满足条件：

$$\varphi(x_i) = y_i \quad (i = 0,1,\cdots,n), \tag{2.1.3}$$

则称 $\varphi(x)$为**插值函数**. 当 $\varphi_i(x)$是代数多项式时，求 $\varphi(x)$仍然是**多项式插值**. 当 $\varphi(x)$是有理函数时称为**有理插值**，当 $\varphi(x)$为三角函数时称为**三角插值**.

插值法是一个古老的课题. 早在公元 6 世纪，我国刘焯已将等距二次插值应用于天文计算. 17 世纪，牛顿和格雷哥里(Gregory)建立了等距节点上的一般插值公式. 18 世纪，拉格朗日(Lagrange)给出了更一般的非等距节点上的插值公式. 由于应用上和理论上的需要，近几十年来，插值法仍不断有新的发展. 特别是利用样条函数的插值，已在精密机械加工，计算机图形学等领域有广泛应用.

2.1.3 插值多项式的存在惟一性

定理 2.1.1 存在惟一的、次数不超过 n 的多项式 p，满足插值条件 $p(x_i)=y_i, i=0,1,\cdots,n$.

此定理有不同的证法.可通过构造多项式证明存在性，利用多项式的次数与根的个数的关系证明惟一性，也可假设

$$p(x)=a_nx^n+a_{n-1}x^{n-1}+\cdots+a_1x+a_0, \qquad (2.1.4)$$

且满足插值条件，即

$$a_nx_i^n+a_{n-1}x_i^{n-1}+\cdots+a_1x_i+a_0=y_i \quad (i=0,1,\cdots,n). \qquad (2.1.5)$$

这是关于 $a_0,a_1,\cdots,a_n$ 的线性代数方程组，因系数行列式不为零，故有惟一解.从而 p 存在且惟一.

插值问题的关键，是怎样具体去求插值多项式.求插值多项式的方法称为**插值法**(method of interpolation).

利用解方程组(2.1.5)去建立形如式(2.1.4)的插值多项式，计算量大，有时还会对精度有较大的影响，因而是不可取的.插值法是一些比较方便的方法，它们对插值多项式的形式作了各种特殊的选取，以便较容易地去具体确定.

2.2 拉格朗日插值

2.2.1 基函数

考虑最简单的插值问题：设离散数据为 $\{(x_k,\delta_{ik})\}_{k=0}^n$，这里 i 是一个非负整数，$0\leqslant i\leqslant n$，δ_{ik} 是克罗内克(Kronecker)符号

$$\delta_{ik}=\begin{cases}1, & k=i,\\ 0, & k\neq i,\end{cases} \qquad (2.2.1)$$

求插值多项式，记这个多项式为 l_i.

由于多项式 l_i 必须满足

$$l_i(x_k) = \delta_{ik} \quad (k = 0, 1, \cdots, n), \tag{2.2.2}$$

即 $x_0, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n$ 是 l_i 的根，因此可取如下形式：

$$l_i(x) = a(x - x_0)\cdots(x - x_{i-1})(x - x_{i+1})\cdots(x - x_n),$$

而且从条件 $l_i(x_i)=1$ 可确定出系数 a，最后得出

$$l_i(x) = \frac{(x - x_0)\cdots(x - x_{i-1})(x - x_{i+1})\cdots(x - x_n)}{(x_i - x_0)\cdots(x_i - x_{i-1})(x_i - x_{i+1})\cdots(x_i - x_n)}$$

$$= \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j}. \tag{2.2.3}$$

$n+1$ 个 n 次多项式 $l_0, l_1, \cdots, l_n$ 称为多项式插值的以 x_0, $x_1, \cdots, x_n$ 为节点的 n 次**基函数**(basis function).

$n=1$ 时的基函数及图形(见图 2.1)：

$$l_0(x) = \frac{x - x_1}{x_0 - x_1}, \quad l_1(x) = \frac{x - x_0}{x_1 - x_0}. \tag{2.2.4}$$

$n=2$ 时的基函数及图形(见图 2.2)：

$$\begin{cases} l_0(x) = \dfrac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}, \\ l_1(x) = \dfrac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)}, \\ l_2(x) = \dfrac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}. \end{cases} \tag{2.2.5}$$

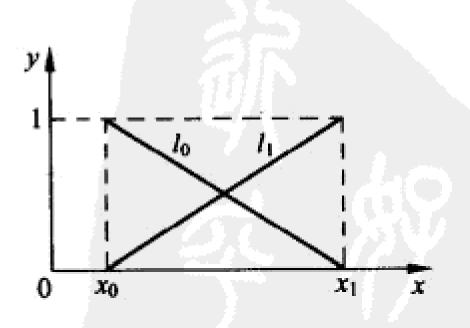


图 2.1

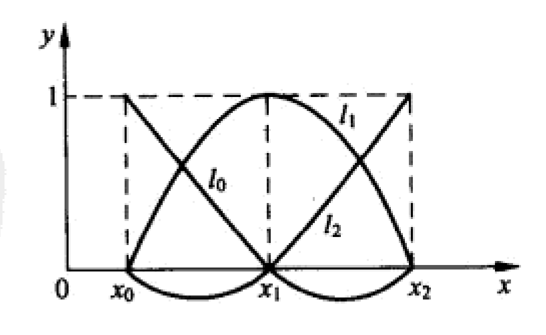


图 2.2

2.2.2 拉格朗日插值多项式

一般离散数据 $\{(x_i, y_i)\}_{i=0}^{n}$ 的插值多项式 L_n 是基函数 l_0, $l_1, \cdots, l_n$ 的线性组合：

$$L_n(x) = \sum_{i=0}^{n} y_i l_i(x)$$

$$= \sum_{i=0}^{n} y_i \frac{(x-x_0)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_n)}{(x_i-x_0)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)}. \tag{2.2.6}$$

显然，L_n 满足插值条件，即

$$L_n(x_j) = y_j \quad (j = 0, 1, \cdots, n).$$

L_n 称为拉格朗日插值多项式. 记

$$W_{n+1}(x) = \prod_{i=0}^{n}(x - x_i)$$

$$= (x-x_0)(x-x_1)\cdots(x-x_n), \tag{2.2.7}$$

于是式(2.2.6)可改写成

$$L_n(x) = W_{n+1}(x)\sum_{i=0}^{n}\frac{y_i}{(x-x_i)W'_{n+1}(x_i)}$$

$$(x \neq x_i, i = 0, 1, \cdots, n). \tag{2.2.8}$$

例 2.2.1 设有离散数据

$$(1,0), (2,-5), (3,-6), (4,3),$$

则插值多项式为

$$L_3(x) = 0 \times \frac{(x-2)(x-3)(x-4)}{(1-2)(1-3)(1-4)} +$$

$$(-5) \times \frac{(x-1)(x-3)(x-4)}{(2-1)(2-3)(2-4)} +$$

$$(-6) \times \frac{(x-1)(x-2)(x-4)}{(3-1)(3-2)(3-4)} +$$

$$3\times\frac{(x-1)(x-2)(x-3)}{(4-1)(4-2)(4-3)}$$
$$=x^3-4x^2+3.$$

2.2.3 余项

设在区间$[a,b]$上用插值多项式 L_n 逼近函数 f，节点 x_0，x_1，…，$x_n\in[a,b]$，且 $f(x_i)=y_i(i=0,1,\cdots,n)$，则其截断误差 R_n 是$[a,b]$上的函数：

$$R_n(x)=f(x)-L_n(x).$$

R_n 也称为 L_n 的**余项**(remainder term). 这时，有下面的余项估计定理.

定理 2.2.2 若 f 在$[a,b]$上存在 $n+1$ 阶导数，则对任何 $x\in[a,b]$有

$$\begin{aligned}R_n(x)&=f(x)-L_n(x)\\&=\frac{f^{(n+1)}(\xi)}{(n+1)!}W_{n+1}(x),\end{aligned}\tag{2.2.9}$$

其中 $\xi\in[a,b]$且依赖于 x，W_{n+1} 由式(2.2.7)定义.

ξ 一般无法具体确定，因此式(2.2.9)只是一个估计式. 如果 $f^{(n+1)}$ 在$[a,b]$上有界(或连续)，即存在常数 $M_{n+1}>0$(连续时可取 $M_{n+1}=\max\limits_{a\leqslant x\leqslant b}|f^{(n+1)}(x)|$)，使

$$|f^{(n+1)}(x)|\leqslant M_{n+1},\quad\forall x\in[a,b].$$

则有余项估计

$$|R_n(x)|\leqslant\frac{M_{n+1}}{(n+1)!}|W_{n+1}(x)|.\tag{2.2.10}$$

对于 $n=1$ 的情况，L_1 称为**线性插值多项式**. 设节点 $x_0<x_1$，函数 f 在区间$[x_0,x_1]$上有连续二阶导数，$y_0=f(x_0)$，$y_1=f(x_1)$. 在$[x_0,x_1]$上用 L_1 逼近 f，有

$$L_1(x) = y_0 \frac{x - x_1}{x_0 - x_1} + y_1 \frac{x - x_0}{x_1 - x_0}, \qquad (2.2.11)$$

其余项为

$$\begin{aligned} R_1(x) &= \frac{1}{2} f''(\xi) W_2(x) \\ &= \frac{1}{2} f''(\xi)(x - x_0)(x - x_1), \quad \xi \in [x_0, x_1]. \end{aligned}$$

由于 $|W_2(x)|$ 在 $x = \frac{x_0 + x_1}{2}$ 达最大值,可得余项的一个界

$$\begin{aligned} |R_1(x)| &\leqslant \frac{1}{2} M_2 \left| W_2\left(\frac{x_0 + x_1}{2}\right) \right| \\ &= \frac{M_2 (x_1 - x_0)^2}{8}, \qquad (2.2.12) \end{aligned}$$

其中 $M_2 = \max\limits_{x_0 \leqslant x \leqslant x_1} |f''(x)|$.

若是为了建立插值多项式,则只要知道函数 f 在节点(离散点)上的值就足够了;但如果要给出余项估计,则要求 f 在一个区间上不仅有明确的表达式,而且还存在高阶导数.

2.3 艾特肯法

2.3.1 问题的提出

拉格朗日插值多项式的优点是简单、易于建立.缺点是增加新节点时原有多项式的计算结果不能利用,必须重新建立;而且其形式不易简化,计算工作量较大.

为了克服上述两个缺点,产生了一些新的途径.艾特肯(Aitken)法利用一列拉格朗日插值多项式,避免了增加新节点时从头开始计算.牛顿插值则从改进插值多项式的形式入手,以便于增加节点和节省计算工作量.

2.3.2 艾特肯法的描述

用 $p_{n_0,n_1,\cdots,n_k}$ 表示节点为 $x_{n_0},x_{n_1},\cdots,x_{n_k}$ 的拉格朗日插值多项式，p_{n_i} 是零次多项式即为常数，

$$p_{n_i}(x)=f(x_{n_i}), \tag{2.3.1}$$

这里 $n_0,n_1,\cdots,n_k$ 是非负整数，f 是被逼近的函数.

容易证明(只要验证满足插值条件)：

$$\begin{aligned}p_{0,1,\cdots,k,n}(x)&=\frac{1}{x_n-x_k}\begin{vmatrix}p_{0,1,\cdots,k}(x) & x_k-x\\ p_{0,1,\cdots,k-1,n}(x) & x_n-x\end{vmatrix}\\ &=p_{0,1,\cdots,k}(x)\frac{x-x_n}{x_k-x_n}+\\ &\qquad p_{0,1,\cdots,k-1,n}(x)\frac{x-x_k}{x_n-x_k}.\end{aligned} \tag{2.3.2}$$

公式(2.3.2)表明，$k+1$ 次插值多项式可由两个 k 次插值多项式通过线性插值得到. 因此，利用这个公式，可以一行接一行地构造表 2.1. 例如：

第 2 行中

$$p_{0,1}(x)=\frac{1}{x_1-x_0}\begin{vmatrix}p_0(x) & x_0-x\\ p_1(x) & x_1-x\end{vmatrix},$$

$$p_{0,2}(x)=\frac{1}{x_2-x_0}\begin{vmatrix}p_0(x) & x_0-x\\ p_2(x) & x_2-x\end{vmatrix}.$$

第 3 行中

$$p_{0,1,2}(x)=\frac{1}{x_2-x_1}\begin{vmatrix}p_{0,1}(x) & x_1-x\\ p_{0,2}(x) & x_2-x\end{vmatrix},$$

等等. 表中第 $n+1$ 行的最后一个多项式即为 n 次插值多项式. 整个计算过程称为**艾特肯法**.

表 2.1

x_i	已知数据	按式(2.3.2)构造多项式			
x_0	$p_0(x)=f(x_0)$				
x_1	$p_1(x)=f(x_1)$	$p_{0,1}(x)$			
x_2	$p_2(x)=f(x_2)$	$p_{0,2}(x)$	$p_{0,1,2}(x)$		
x_3	$p_3(x)=f(x_3)$	$p_{0,3}(x)$	$p_{0,1,3}(x)$	$p_{0,1,2,3}(x)$	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$
x_n	$p_n(x)=f(x_n)$	$p_{0,n}(x)$	$p_{0,1,n}(x)$	$p_{0,1,2,n}(x)\cdots$	$p_{0,1,\cdots,n}(x)$

利用艾特肯法，每增加一个节点，只要在表 2.1 中多计算一行即可.

2.3.3 计算工作量

下面比较在求 n 次插值多项式的值时，拉格朗日插值与艾特肯法的运算次数.

通过计算，先将表达式(2.2.6)化为

$$L_n(x)=\sum_{i=0}^{n}\left\{a_i\prod_{\substack{j=0\\j\neq i}}^{n}(x-x_j)\right\}, \tag{2.3.3}$$

计算

$$a_i=y_i\Big/\prod_{\substack{j=0\\j\neq i}}^{n}(x_i-x_j)\quad(i=0,1,\cdots,n)$$

需 n^2-1 次乘法，$n+1$ 次除法，n^2+n 次减法. 然后利用(2.3.3)式求值，每次求值需 n^2+n 次乘法，n^2+n 次减法，n 次加法. 如果仅求一个值，拉格朗日插值的计算工作量总共为 $2n^2+n-1$ 次乘法，$n+1$ 次除法，$2n^2+2n$ 次减法及 n 次加法.

艾特肯法求 n 次插值多项式的值可以直接从表 2.1 出发，表中共有 $n(n+1)/2$ 项，每项按公式(2.3.2)求值需 2 次乘法，1 次

除法及 4 次减法，因此求 n 次多项式的一个值共需 n^2+n 次乘法，$n(n+1)/2$ 次除法，$2n^2+2n$ 次减法.

例 2.3.1 已知 $f(x)=\sinh x$ 的离散数据

x_i	0.00	0.20	0.30	0.50
$f(x_i)$	0.00000	0.20134	0.30452	0.52110

利用三次插值求 $f(0.23)$ 的近似值.

解 采用拉格朗日插值

$$f(0.23)\approx 0.20134\times\frac{0.23\times(-0.07)\times(-0.27)}{0.20\times(-0.10)\times(-0.30)}+$$

$$0.30452\times\frac{0.23\times 0.03\times(-0.27)}{0.30\times 0.10\times(-0.20)}+$$

$$0.52110\times\frac{0.23\times 0.03\times(-0.07)}{0.50\times 0.30\times 0.20}$$

$$\approx 0.232035.$$

采用艾特肯法列表如下：

0.00	0.00000				−23
0.20	0.20134	0.231541			−3
0.30	0.30452	0.233465	0.232118		7
0.50	0.52110	0.239706	0.232358	<u>0.232034</u>	27

如果增加一个节点，例如 $x_4=0.60$，其函数值 $f(0.60)=0.63665$，并改用四次插值计算 $f(0.23)$ 的近似值. 则拉格朗日插值必须从头计算，而艾特肯法只增加如下一行：

0.60	0.63665	0.244049	0.232479	0.232034	<u>0.232034</u>	37

2.4 均差与牛顿插值

2.4.1 均差

设函数 f 在 $n+1$ 个节点 $x_0,x_1,\cdots,x_n$ 上的值 $f(x_0),f(x_1),\cdots,f(x_n)$为已知.牛顿插值是把插值多项式表示成如下形式：

$$N_n(x)=a_0+a_1(x-x_0)+a_2(x-x_0)(x-x_1)+\cdots+a_n(x-x_0)(x-x_1)\cdots(x-x_{n-1}). \tag{2.4.1}$$

这种形式显然便于求值,而且增加一个新节点 x_{n+1} 时,只须增加一个新项

$$a_{n+1}(x-x_0)(x-x_1)\cdots(x-x_n),$$

根据插值条件

$$N_n(x_i)=f(x_i)\quad(i=0,1,\cdots,n),$$

可以逐个确定出系数

$$a_0=f(x_0),$$

$$a_1=\frac{f(x_1)-f(x_0)}{x_1-x_0},$$

$$a_2=\frac{\dfrac{f(x_2)-f(x_0)}{x_2-x_0}-\dfrac{f(x_1)-f(x_0)}{x_1-x_0}}{x_2-x_1},$$

$$\vdots$$

为了得到系数 a_i 的一般表达式,有如下均差的定义.

定义 2.4.1 记 $f[x_m]=f(x_m)$,称为**零阶均差**.零阶均差的差商

$$f[x_0,x_m]=\frac{f(x_m)-f(x_0)}{x_m-x_0},$$

称为函数 f 关于 x_0,x_m 的**一阶均差**.一阶均差的差商

$$f[x_0, x_1, x_m] = \frac{f[x_0, x_m] - f[x_0, x_1]}{x_m - x_1},$$

称为 f 的**二阶均差**.一般地,$k-1$ 阶均差的差商

$$f[x_0, x_1, \cdots, x_{k-1}, x_m] = \frac{f[x_0, \cdots, x_{k-2}, x_m] - f[x_0, x_1, \cdots, x_{k-1}]}{x_m - x_{k-1}}, \quad (2.4.2)$$

称为 f 的 **k 阶均差**(k-th divided difference).

均差有如下基本性质:

(1) 均差与函数值的关系为

$$f[x_0, x_1, \cdots, x_n] = \sum_{j=0}^{n} \frac{f(x_j)}{(x_j - x_0)\cdots(x_j - x_{j-1})(x_j - x_{j+1})\cdots(x_j - x_n)}. \quad (2.4.3)$$

可用归纳法给以证明.

(2) 均差关于所含节点是对称的,即有

$$f[x_0, x_1, \cdots, x_n] = f[x_1, x_0, x_2, \cdots, x_n] = \cdots = f[x_1, \cdots, x_n, x_0].$$

这是(1)的直接推论.

(3) 由(2)及式(2.4.2)可得

$$f[x_0, x_1, \cdots, x_{k-1}, x_m] = \frac{f[x_1, \cdots, x_{k-1}, x_m] - f[x_0, x_1, \cdots, x_{k-1}]}{x_m - x_0}. \quad (2.4.4)$$

(4) 设 f 在 $[a,b]$ 上存在 n 阶导数,且 $x_0, x_1, \cdots, x_n \in [a,b]$,则

$$f[x_0, x_1, \cdots, x_n] = \frac{f^{(n)}(\xi)}{n!}, \quad (2.4.5)$$

其中 $\xi \in [a,b]$.

牛顿均差插值多项式为

$$N_3(x)=1.0067x+0.08367x(x-0.20)+0.17332x(x-0.20)(x-0.30),$$

由此可算出

$$f(0.23)\approx N_3(0.23)\approx 0.23203.$$

因 $f(x)=\sinh x, f^{(4)}(x)=\sinh x$,余项为

$$R_3(x)=\frac{1}{4!}x(x-0.20)(x-0.30)(x-0.50)\sinh\xi,\quad 0<\xi<0.5,$$

可得误差估计

$$|R_3(0.23)|<\frac{1}{24}\times 0.23\times 0.03\times 0.07\times 0.27\times 0.53\approx 0.000003,$$

如果增加一个节点 $x_4=0.60$,则上述均差表增加如下一行:

0.60	0.63665	1.0611	0.13596	0.17429	<u>0.00974</u>

插值多项式为

$$N_4(x)=N_3(x)+0.00974x(x-0.20)(x-0.30)(x-0.50).$$

2.5 差分与等距节点插值

2.5.1 差分

前面的插值公式一般只应用于非等距节点(即节点不是等距离分布)的情形.在实际应用中,常采用等距节点

$$x_i=x_0+ih\quad(i=0,\pm 1,\pm 2,\cdots),$$

其中 h 是常数,称为**步长**(step width).这时利用差分可以导出计算上更为有效的插值多项式.

2.4.2 牛顿插值公式

利用式(2.4.4),可以得到

$$f(x) = f(x_0) + f[x,x_0](x - x_0),$$

$$f[x,x_0] = f[x_0,x_1] + f[x,x_0,x_1](x - x_1),$$

$$f[x,x_0,x_1] = f[x_0,x_1,x_2] + f[x,x_0,x_1,x_2](x - x_2),$$

$$\vdots$$

$$f[x,x_0,\cdots,x_{n-1}] = f[x_0,x_1,\cdots,x_n] + f[x,x_0,\cdots,x_n](x - x_n).$$

依次将后一式代入前一式,最后有

$$f(x) = N_n(x) + R_n(x), \tag{2.4.6}$$

其中

$$\begin{aligned} N_n(x) = &f(x_0) + f[x_0,x_1](x - x_0) + \\ &f[x_0,x_1,x_2](x - x_0)(x - x_1) + \cdots + \\ &f[x_0,x_1,\cdots,x_n](x - x_0)(x - x_1)\cdots(x - x_{n-1}), \end{aligned} \tag{2.4.7}$$

$$\begin{aligned} R_n(x) &= f(x) - N_n(x) \\ &= f[x,x_0,\cdots,x_n]W_{n+1}(x), \end{aligned} \tag{2.4.8}$$

W_{n+1} 由式(2.2.7)定义.

由式(2.4.7)确定的多项式 N_n 显然满足插值条件,次数不超过 n,且具有式(2.4.1)的形式.因此,式(2.4.1)中的系数为

$$a_i = f[x_0,x_1,\cdots,x_i] \quad (i = 0,1,\cdots,n).$$

多项式 N_n 称为**牛顿均差插值多项式**.根据插值多项式的惟一性,N_n 与拉格朗日插值多项式 L_n 只是形式上不同.

当 f 存在 $n+1$ 阶导数时,公式(2.4.8)给出均差形式的余项 R_n 与公式(2.2.9)等价,并由此可推出均差与导数的关系式(2.4.5).但公式(2.4.8)更有一般性,它在 f 是由离散数据确定或导数不存在的情形下仍有意义.

用 f_i 表示函数 f 在节点 x_i 的值，即

$$f_i = f(x_i) \quad (i = 0, \pm 1, \pm 2, \cdots).$$

定义 2.5.1 令

$$\Delta f_i = f_{i+1} - f_i,$$
$$\nabla f_i = f_i - f_{i-1},$$
$$\delta f_i = f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}},$$

这里 $f_{i+\frac{1}{2}} = f\left(x_i + \frac{h}{2}\right)$，$f_{i-\frac{1}{2}} = f\left(x_i - \frac{h}{2}\right)$. Δf_i，∇f_i，δf_i 分别称为 f 在 x_i 的**一阶向前差分**(first forward difference)，**一阶向后差分**(first backward difference)，**一阶中心差分**(first central difference). Δ，∇，δ 称为(向前，向后，中心)**差分算子**(difference operator).

由此，可递推地定义 n 阶差分为

$$\Delta^n f_i = \Delta^{n-1} f_{i+1} - \Delta^{n-1} f_i,$$
$$\nabla^n f_i = \nabla^{n-1} f_i - \nabla^{n-1} f_{i-1},$$
$$\delta^n f_i = \delta^{n-1} f_{i+\frac{1}{2}} - \delta^{n-1} f_{i-\frac{1}{2}},$$

并规定**零阶差分**为

$$\Delta^0 f_i = \nabla^0 f_i = \delta^0 f_i = f_i$$

例 2.5.2 二阶差分和三阶差分

$$\Delta^2 f_i = \Delta(\Delta f_i) = \Delta f_{i+1} - \Delta f_i = f_{i+2} - 2f_{i+1} + f_i,$$

$$\Delta^3 f_i = \Delta^2 f_{i+1} - \Delta^2 f_i = f_{i+3} - 3f_{i+2} + 3f_{i+1} - f_i,$$

$$\nabla^2 f_i = \nabla(\nabla f_i) = \nabla f_i - \nabla f_{i-1} = f_i - 2f_{i-1} + f_{i-2},$$

$$\nabla^3 f_i = \nabla^2 f_i - \nabla^2 f_{i-1} = f_i - 3f_{i-1} + 3f_{i-2} - f_{i-3},$$

$$\delta^2 f_i = \delta(\delta f_i) = \delta f_{i+\frac{1}{2}} - \delta f_{i-\frac{1}{2}} = f_{i+1} - 2f_i + f_{i-1},$$

$$\delta^3 f_i = \delta^2 f_{i+\frac{1}{2}} - \delta^2 f_{i-\frac{1}{2}} = f_{i+\frac{3}{2}} - 3f_{i+\frac{1}{2}} + 3f_{i-\frac{1}{2}} - f_{i-\frac{3}{2}}.$$

2.4.3 计算工作量

计算均差可按表 2.2 逐行地进行，表中加横线的各阶均差就是牛顿插值多项式的系数.

表 2.2

x_i	$f(x_i)$	一阶均差	二阶均差	三阶均差	…	n 阶均差
x_0	$\underline{f(x_0)}$					
x_1	$f(x_1)$	$\underline{f[x_0,x_1]}$				
x_2	$f(x_2)$	$f[x_0,x_2]$	$\underline{f[x_0,x_1,x_2]}$			
x_3	$f(x_3)$	$f[x_0,x_3]$	$f[x_0,x_1,x_3]$	$\underline{f[x_0,x_1,x_2,x_3]}$		
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	
x_n	$f(x_n)$	$f[x_0,x_n]$	$f[x_0,x_1,x_n]$	$f[x_0,x_1,x_2,x_n]$	…	$\underline{f[x_0,x_1,\cdots,x_n]}$

为了建立 n 次牛顿插值多项式，求各阶均差需进行 $\frac{1}{2}(n^2+n)$ 次除法和 n^2+n 次减法. 得出公式(2.4.7)后，可利用 Horner 算法(1.4.5)求值，每次求值需 n 次乘法、n 次减法和 n 次加法. 与拉格朗日插值相比，牛顿插值大大节省了工作量.

例 2.4.2 考虑例 2.3.1 中的问题.

解 利用牛顿插值，均差表如下：

0.00	$\underline{0.00000}$			
0.20	0.20134	$\underline{1.0067}$		
0.30	0.30452	1.0151	$\underline{0.08367}$	
0.50	0.52110	1.0422	0.11833	$\underline{0.17332}$

定理 2.5.3 各阶差分可用函数值表示如下：

$$\Delta^n f_i = \sum_{k=0}^{n} (-1)^k \binom{n}{k} f_{n+i-k},$$

$$\nabla^n f_i = \sum_{k=0}^{n} (-1)^k \binom{n}{k} f_{i-k},$$

$$\delta^n f_i = \sum_{k=0}^{n} (-1)^k \binom{n}{k} f_{\frac{n}{2}+i-k},$$

其中$\binom{n}{k}=\frac{n(n-1)\cdots(n-k+1)}{k!}$是二项式展开系数.

定理 2.5.4 函数值可用各阶差分表示如下：

$$f_{i+n} = \sum_{k=0}^{n} \binom{n}{k} \Delta^k f_i,$$

$$f_{i-n} = \sum_{k=0}^{n} (-1)^k \binom{n}{k} \nabla^k f_i,$$

$$f_{i+n} = \sum_{k=0}^{n} \binom{n}{k} \delta^k f_{i+\frac{k}{2}}.$$

定理 2.5.5 差分与均差的关系如下：

$$\Delta^m f_i = m! h^m f[x_i, x_{i+1}, \cdots, x_{i+m}],$$

$$\nabla^m f_i = m! h^m f[x_{i-m}, x_{i-m+1}, \cdots, x_i],$$

$$\delta^{2m+1} f_{i+\frac{1}{2}} = (2m+1)! h^{2m+1} f[x_{i-m}, x_{i-m+1}, \cdots, x_{i+m+1}],$$

$$\delta^{2m} f_i = (2m)! h^{2m} f[x_{i-m}, x_{i-m+1}, \cdots, x_{i+m}].$$

以上三个定理均可用数学归纳法证明.由式(2.4.5)及定理 2.5.5 可得差分和导数的关系,例如向前差分与导数的关系为

$$\Delta^n f_i = h^n f^{(n)}(\xi), \quad \xi \in [x_i, x_{i+n}]. \tag{2.5.1}$$

利用函数值 $f_i(i=0,\pm 1,\pm 2,\cdots)$计算各阶差分特别简单,仅包含减法运算,可以构造差分表 2.3.

表 2.3

f_i	$\Delta(\nabla,\delta)$	$\Delta^2(\nabla^2,\delta^2)$	$\Delta^3(\nabla^3,\delta^3)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
f_{-1}	$\Delta f_{-1}(\nabla f_0,\delta f_{-1/2})$		
f_0	$\Delta f_0(\nabla f_1,\delta f_{1/2})$	$\Delta^2 f_{-1}(\nabla^2 f_1,\delta^2 f_0)$	$\Delta^3 f_{-1}(\nabla^3 f_2,\delta^3 f_{1/2})$
f_1	$\Delta f_1(\nabla f_2,\delta f_{3/2})$	$\Delta^2 f_0(\nabla^2 f_2,\delta^2 f_1)$	
f_2	$\vdots$	$\vdots$	$\vdots$
$\vdots$			

2.5.2 牛顿差分插值公式

利用定理 2.5.5,在牛顿均差插值公式(2.4.7)中将均差替换为差分,可以得出不同形式的插值公式.

在以 $x_i, x_{i+1}, \cdots, x_{i+n}$ 为节点的牛顿均差插值公式中,用向前差分代替均差,并令 $x=x_i+th$, $0\leqslant t\leqslant n$,可得

$$N_n(x_i+th)=f_i+\binom{t}{1}\Delta f_i+\binom{t}{2}\Delta^2 f_i+\cdots+\binom{t}{n}\Delta^n f_i \tag{2.5.2}$$

此即**牛顿向前差分公式**,式中

$$\binom{t}{k}=\frac{t(t-1)\cdots(t-k+1)}{k!},$$

t 是变量.这时

$$W_{n+1}(x)=\prod_{k=0}^{n}(x-x_{i+k})=t(t-1)\cdots(t-n)h^{n+1}.$$

根据式(2.2.9),余项可以写成

$$R_n(x_i+th)=\binom{t}{n+1}h^{n+1}f^{(n+1)}(\xi),$$
$$\xi\in[x_i,x_{i+n}]. \tag{2.5.3}$$

类似地,在以 $x_i, x_{i-1}, \cdots, x_{i-n}$(从大到小排列)为节点的牛顿均

差插值公式中,用向后差分代替均差,并令 $x=x_i+th$, $-n\leqslant t\leqslant 0$,可得

$$N_n(x_i+th)=f_i+\binom{t}{1}\nabla f_i+\binom{t+1}{2}\nabla^2 f_i+\cdots+\binom{t+n-1}{n}\nabla^n f_i, \tag{2.5.4}$$

称为**牛顿向后差分公式**.余项可写成

$$R_n(x_i+th)=\binom{t+n}{n+1}h^{n+1}f^{(n+1)}(\xi),\quad \xi\in[x_{i-n},x_i]. \tag{2.5.5}$$

2.5.3 高斯公式

在 $n+1$ 个节点按 $x_i, x_{i+1}, x_{i-1}, x_{i+2}, x_{i-2}, \cdots$ 顺序排列的牛顿均差插值公式中,根据定理 2.5.5,用中心差商代替均差,并令 $x=x_i+th$,可得另一种插值公式.当 $n=2m$ 时,节点为 $x_i, x_{i+1}, x_{i-1}, \cdots, x_{i+m}, x_{i-m}$ $(-m\leqslant t\leqslant m)$,

$$\begin{aligned}G_n(t)&=N_n(x_i+th)\\&=f_i+\binom{t}{1}\delta f_{i+\frac{1}{2}}+\binom{t}{2}\delta^2 f_i+\binom{t+1}{3}\delta^3 f_{i+\frac{1}{2}}+\\&\quad\binom{t+1}{4}\delta^4 f_i+\cdots+\binom{t+m-1}{2m-1}\delta^{2m-1}f_{i+\frac{1}{2}}+\\&\quad\binom{t+m-1}{2m}\delta^{2m}f_i,\end{aligned} \tag{2.5.6}$$

余项可写成

$$R_n(x_i+th)=\binom{t+m}{2m+1}h^{2m+1}f^{(2m+1)}(\xi),\quad \xi\in[x_{i-m},x_{i+m}]. \tag{2.5.7}$$

当 $n=2m+1$ 时,增加一个节点 x_{i+m+1}, $-m\leqslant t\leqslant m+1$,只需在式(2.5.6)中加一项:

$$G_n(t)=f_i+\binom{t}{1}\delta f_{i+\frac{1}{2}}+\binom{t}{2}\delta^2 f_i+\binom{t+1}{3}\delta^3 f_{i+\frac{1}{2}}+$$
$$\binom{t+1}{4}\delta^4 f_i+\cdots+\binom{t+m-1}{2m-1}\delta^{2m-1} f_{i+\frac{1}{2}}+$$
$$\binom{t+m-1}{2m}\delta^{2m} f_i+\binom{t+m}{2m+1}\delta^{2m+1} f_{i+\frac{1}{2}},\tag{2.5.8}$$

余项变为

$$R_n(x_i+th)=\binom{t+m}{2m+2}h^{2m+2}f^{(2m+2)}(\xi),$$
$$\xi\in[x_{i-m},x_{i+m+1}].\tag{2.5.9}$$

式(2.5.6)与式(2.5.8)称为**高斯向前差分公式**.

类似地,如果 $n+1$ 个节点改为按 $x_i,x_{i-1},x_{i+1},x_{i-2},x_{i+2},\cdots$ 顺序排列,$x=x_i+th$,则当 $n=2m$ 时,$-m\leqslant t\leqslant m$,有

$$G_n(t)=f_i+\binom{t}{1}\delta f_{i-\frac{1}{2}}+\binom{t+1}{2}\delta^2 f_i+\binom{t+1}{3}\delta^3 f_{i-\frac{1}{2}}+$$
$$\binom{t+2}{4}\delta^4 f_i+\cdots+\binom{t+m-1}{2m-1}\delta^{2m-1} f_{i-\frac{1}{2}}+$$
$$\binom{t+m}{2m}\delta^{2m} f_i,\tag{2.5.10}$$

余项为

$$R_n(x_i+th)=\binom{t+m}{2m+1}h^{2m+1}f^{(2m+1)}(\xi),$$
$$\xi\in[x_{i-m},x_{i+m}].\tag{2.5.11}$$

当 $n=2m+1$ 时,$-m-1\leqslant t\leqslant m$,有

$$G_n(t)=f_i+\binom{t}{1}\delta f_{i-\frac{1}{2}}+\binom{t+1}{2}\delta^2 f_i+\binom{t+1}{3}\delta^3 f_{i-\frac{1}{2}}+$$
$$\binom{t+2}{4}\delta^4 f_i+\cdots+\binom{t+m-1}{2m-1}\delta^{2m-1} f_{i-\frac{1}{2}}+$$
$$\binom{t+m}{2m}\delta^{2m} f_i+\binom{t+m}{2m+1}\delta^{2m+1} f_{i-\frac{1}{2}},\tag{2.5.12}$$

余项为

$$R_n(x_i+th)=\binom{t+m+1}{2m+2}h^{2m+2}f^{(2m+2)}(\xi),$$

$$\xi\in[x_{i-m-1},x_{i+m}]. \tag{2.5.13}$$

式(2.5.10)与式(2.5.12)称为**高斯向后公式.**

2.5.4 斯特林公式

取高斯向前公式(2.5.6)与高斯向后公式(2.5.10)的平均,即可得到以 $x_{i-m},\cdots,x_i,\cdots,x_{i+m}$ 为节点的插值公式

$$\begin{aligned}S_n(t)=f_i&+\frac{1}{2}\binom{t}{1}(\delta f_{i+\frac{1}{2}}+\delta f_{i-\frac{1}{2}})+\\&\frac{1}{2}\left[\binom{t}{2}+\binom{t+1}{2}\right]\delta^2 f_i+\\&\frac{1}{2}\binom{t+1}{3}(\delta^3 f_{i+\frac{1}{2}}+\delta^3 f_{i-\frac{1}{2}})+\cdots+\\&\frac{1}{2}\left[\binom{t+m-1}{2m}+\binom{t+m}{2m}\right]\delta^{2m}f_i,\end{aligned} \tag{2.5.14}$$

这里 $-m\leqslant t=\frac{x-x_i}{h}\leqslant m,n=2m$. 余项为

$$R_n(x_i+th)=\binom{t+m}{2m+1}h^{2m+1}f^{(2m+1)}(\xi),$$

$$\xi\in[x_{i-m},x_{i+m}]. \tag{2.5.15}$$

式(2.5.14)称为**斯特林(Stirling)公式**. 这个公式用的节点个数是奇数.

2.5.5 贝塞尔公式

在高斯向后公式(2.5.12)中用 $i+1$ 代替 i,同时用 $t-1$ 代替 t,然后与高斯向前公式(2.5.8)取平均,可得到以 $x_{i-m},\cdots,x_i,\cdots,$

x_{i+m}, x_{i+m+1} 为节点的另一个插值公式

$$B_n(t)=\frac{1}{2}(f_i+f_{i+1})+\frac{1}{2}\left[\binom{t}{1}+\binom{t-1}{1}\right]\delta f_{i+\frac{1}{2}}+$$

$$\frac{1}{2}\binom{t}{2}(\delta^2 f_i+\delta^2 f_{i+1})+$$

$$\frac{1}{2}\left[\binom{t+1}{3}+\binom{t}{3}\right]\delta^3 f_{i+\frac{1}{2}}+\cdots+$$

$$\frac{1}{2}\left[\binom{t+m}{2m+1}+\binom{t+m-1}{2m+1}\right]\delta^{2m+1} f_{i+\frac{1}{2}}, \quad (2.5.16)$$

这里 $-m\leqslant t=\frac{x-x_i}{h}\leqslant m+1, n=2m+1$. 余项为

$$R_n(x_i+th)=\binom{t+m}{2m+2}h^{2m+2}f^{(2m+2)}(\xi),$$

$$\xi\in[x_{i-m}, x_{i+m+1}]. \quad (2.5.17)$$

式(2.5.16)称为**贝塞尔(Bessel)公式**. 这个公式只能用于偶数个节点.

2.5.6 等距节点插值公式的使用

在等距节点的情形下,利用差分建立的各种插值公式可以节省大量的计算工作量. 计算 1 至 n 阶差分总共只需进行 $n(n+1)/2$ 次减法. 概括地说,它们都是用来产生和分析逼近于被插函数的多项式序列. 牛顿向前差分与向后差分公式分别适用于新节点总是等距地沿增大的方向引入和总是等距地沿变小的方向引入,或者分别适用于函数的离散数据表的表首附近的计算和表末附近的计算. 高斯,斯特林和贝塞尔等中心差分的公式适用于在一节点两边,新的节点等距交替地引入,或者适用于在离散数据表的中间附近的计算.

以 $f(x)=\sin x$ 为例,说明等距节点各种插值公式的应用.

表 2.4 是经计算得到的一个差分表：

表 2.4

x	$f(x)$	$\Delta(\nabla,\delta)$	$\Delta^2(\nabla^2,\delta^2)$	$\Delta^3(\nabla^3,\delta^3)$	$\Delta^4(\nabla^4,\delta^4)$	$\Delta^5(\nabla^5,\delta^5)$
1.0	0.84147					
		0.04974				
1.1	0.89121		−0.00891			
		0.04083		−0.00040		
1.2	0.93204		−0.00931		0.00008	
		0.03152		−0.00032		0.00002
1.3	0.96356		−0.00963		0.00010	
		0.02189		−0.00022		0.00001
1.4	0.98545		−0.00985		0.00011	
		0.01204		−0.00011		−0.00003
1.5	0.99749		−0.00996		0.00008	
		0.00208		−0.00003		0.00004
1.6	0.99957		−0.00999		0.00012	
		−0.00791		0.00009		
1.7	0.99166		−0.00990			
		−0.01781				
1.8	0.97385					

检验差分表的正确性可根据每列差分之和应等于前一列的最后一数与最前一数之差. 上述表中五阶差分已近于零,因此只能用至四阶差分.

为了求 $f(1.02)$ 的近似值,使用牛顿向前差分公式

$$N_4(1+0.1t)=0.84147+0.04974t-0.00891\frac{t(t-1)}{2!}-0.00040\frac{t(t-1)(t-2)}{3!}+0.00008\frac{t(t-1)(t-2)(t-3)}{4!},$$

取 $t=0.2$,得

$$f(1.02)\approx N_4(1.02)\approx 0.85211.$$

为了求 $f(1.75)$ 的近似值,使用牛顿向后差分公式

$$N_4(1.8+0.1t)=0.97385-0.01781t-0.00990\frac{t(t+1)}{2!}+$$

$$0.00009\frac{t(t+1)(t+2)}{3!}+$$

$$0.00012\frac{t(t+1)(t+2)(t+3)}{4!},$$

取 $t=-0.5$，得

$$f(1.75)\approx N_4(1.75)\approx 0.98398.$$

为了求 $f(1.22)$的近似值，利用中心差分插值公式. 高斯公式总可用斯特林公式或贝塞尔公式等价地代替. 斯特林公式为

$$S_4(t)=0.93204+\frac{1}{2}(0.03152+0.04083)t-$$

$$0.00931\frac{t^2}{2!}-\frac{1}{2}(0.00032+0.00040)\frac{t(t^2-1)}{3!}+$$

$$0.00008\frac{t^2(t^2-1)}{4!},$$

其中 $t=\frac{x-1.2}{0.1}$. 当 $x=1.22$ 时 $t=0.2$，得

$$f(1.22)\approx S_4(0.2)\approx 0.93910.$$

贝塞尔公式为

$$B_4(t)=\frac{1}{2}(0.93204+0.96356)+0.03152\left(t-\frac{1}{2}\right)-$$

$$\frac{1}{2}(-0.00963-0.00931)\frac{t(t-1)}{2!}-$$

$$0.00032\frac{t(t-1)\left(t-\frac{1}{2}\right)}{3!}-\frac{1}{2}(0.00008+$$

$$0.00010)\frac{t(t^2-1)(t-2)}{4!},$$

同样有

$$f(1.22)\approx B_4(0.2)\approx 0.93910.$$

2.6 埃尔米特插值

2.6.1 一般提法

在应用中，不少实际的插值问题不但要求在节点上函数值相等，而且还要求对应的导数值或高阶导数值均相等，满足这种要求的插值多项式就是埃尔米特(Hermite)插值多项式，它的一般提法是：设函数 f 在节点 $x_0, x_1, \cdots, x_n$ 的函数值与导数值为

$$f(x_i) = f_i,\quad f'(x_i) = f'_i,\quad \cdots,\quad f^{(\alpha_i-1)}(x_i) = f_i^{(\alpha_i-1)},$$
$$i = 0,1,\cdots,n,$$

其中 $\alpha_0, \alpha_1, \cdots, \alpha_n$ 是正整数. 寻求一个次数尽可能低的多项式 H，使得满足条件

$$\mathrm{H}^{(k)}(x_i) = f_i^{(k)}\quad (k = 0,1,\cdots,\alpha_i - 1, i = 0,1,\cdots,n). \tag{2.6.1}$$

可以证明，存在惟一的满足插值条件(2.6.1)的次数不超过 $\alpha = \sum_{i=0}^{n}\alpha_i - 1$ 的多项式 H，即所谓埃尔米特**插值多项式**，其形式可写成

$$\mathrm{H}(x) = \sum_{i=0}^{n}\sum_{k=0}^{\alpha_i-1} f_i^{(k)} h_{ik}(x), \tag{2.6.2}$$

其中 h_{ki} 是 α 次多项式，满足条件

$$h_{ik}^{(l)}(x_j) = \begin{cases} 1, & j = i \quad 且 \quad l = k. \\ 0, & 其他. \end{cases}$$

作为一种具体的情形，设

$$f(x_i) = f_i, f'(x_i) = f'_i \quad (i = 0,1,\cdots,n), \tag{2.6.3}$$

要求构造一个次数不超过 $2n+1$ 的多项式，记作 H_{2n+1}，使得

$$H_{2n+1}(x_i) = f_i, \quad H'_{2n+1}(x_i) = f'_i \quad (i = 0,1,\cdots,n). \tag{2.6.4}$$

从几何上看，即要求 H_{2n+1} 与 f 在 $n+1$ 个节点处相切，这时 x_0，$x_1,\cdots,x_n$ 称为**二重节点**.

2.6.2 插值多项式的建立与余项

先考虑特殊情况：求 $2n+1$ 次的多项式 α_i 和 $\beta_i(0\leqslant i\leqslant n)$，满足条件

$$\alpha_i(x_j) = \delta_{ij}, \quad \alpha'_i(x_j) = 0 \quad (j = 0,1,\cdots,n), \tag{2.6.5}$$

$$\beta_i(x_j) = 0, \quad \beta'_i(x_j) = \delta_{ij} \quad (j = 0,1,\cdots,n), \tag{2.6.6}$$

这里 δ_{ij} 是克罗内罗记号，其定义见式(2.2.1). 这是两个最简单的埃尔米特插值问题.

由条件(2.6.5)与(2.6.6)，及多项式的根和重根的性质，可直接确定 β_i 为

$$\beta_i(x) = (x - x_i)l_i^2(x), \tag{2.6.7}$$

并且可确定 α_i 为

$$\alpha_i(x) = [a(x - x_i) + 1]l_i^2(x),$$

其中 a 是待定常数. 以上两式中的 l_i 是由式(2.2.3)定义的拉格朗日插值的基函数. 根据条件 $\alpha'_i(x_i)=0$ 可定出 a，并由此得出

$$\alpha_i(x) = \left[1 - 2(x - x_j)\sum_{\substack{j=0\\j\neq i}}^{n}\frac{1}{x_i - x_j}\right]l_i^2(x), \tag{2.6.8}$$

α_i 与 $\beta_i(i=0,1,\cdots,n)$是关于插值条件(2.6.4)的插值问题的基函数，埃尔米特插值多项式 H_{2n+1} 明显是这些基函数的线性组合：

$$H_{2n+1}(x) = \sum_{i=0}^{n} f_i \alpha_i(x) + \sum_{i=0}^{n} f'_i \beta_i(x). \qquad (2.6.9)$$

设函数 f 在区间$[a,b]$上存在 $2n+2$ 阶导数，$x_0, x_1, \cdots, x_n \in [a,b]$，且用多项式 H_{2n+1} 近似 f，R_{2n+1}是插值余项. 可以证明

$$\begin{aligned} R_{2n+1}(x) &= f(x) - H_{2n+1}(x) \\ &= \frac{f^{(2n+2)}(\xi)}{(2n+2)!} W_{n+1}^2(x), \end{aligned} \qquad (2.6.10)$$

其中 $\xi \in [a,b]$且依赖于 x，W_{n+1}仍由式(2.2.7)定义.

当 $n=1$，节点为 x_0, x_1 时，基函数为

$$\alpha_0(x) = \left(1 + 2\,\frac{x-x_0}{x_1-x_0}\right)\left(\frac{x-x_1}{x_0-x_1}\right)^2,$$

$$\alpha_1(x) = \left(1 + 2\,\frac{x-x_1}{x_0-x_1}\right)\left(\frac{x-x_0}{x_1-x_0}\right)^2,$$

$$\beta_0(x) = (x-x_0)\left(\frac{x-x_1}{x_0-x_1}\right)^2,$$

$$\beta_1(x) = (x-x_1)\left(\frac{x-x_0}{x_1-x_0}\right)^2,$$

插值多项式 H_3 为

$$\begin{aligned} H_3(x) = &\left[f_0 \cdot \left(1 + 2\,\frac{x-x_0}{x_1-x_0}\right) + f'_0 \cdot (x-x_0)\right]\left(\frac{x-x_1}{x_0-x_1}\right)^2 + \\ &\left[f_1 \cdot \left(1 + 2\,\frac{x-x_1}{x_0-x_1}\right) + f'_1 \cdot (x-x_1)\right]\left(\frac{x-x_0}{x_1-x_0}\right)^2, \end{aligned}$$

(2.6.11)

余项 R_3 为

$$R_3(x) = \frac{f^{(4)}(\xi)}{4!}(x-x_0)^2(x-x_1)^2, \qquad (2.6.12)$$

其中 ξ 在 x_0 与 x_1 之间.

2.6.3 重节点均差与均差形式的埃尔米特插值多项式

为简化埃尔米特插值多项式的计算，需引入重节点均差的概念.

由于均差 $f[x_0,x_1,\cdots,x_n]$是 $n+1$ 个变量 $x_0,x_1,\cdots,x_n\in[a,b]$的连续函数，故可借助极限过程来定义自变量重合的均差，即

$$f[x_0,x_0]=\lim_{x_1\to x_0}\frac{f(x_1)-f(x_0)}{x_1-x_0}=f'(x_0),$$

$$f[x_0,x_0,x_1]=\frac{f[x_0,x_1]-f[x_0,x_0]}{x_1-x_0}=\frac{f[x_0,x_1]-f'(x_0)}{x_1-x_0}.$$

一般地，若 $f\in C^n[a,b]$，由式(2.4.5)，令 $x_i\to x_0(i=1,\cdots,n)$，得

$$f[x_0,x_0,\cdots,x_0]=\frac{f^{(n)}(x_0)}{n!}. \qquad (2.6.13)$$

因此，牛顿插值多项式也可用于埃尔米特插值. 若考虑 f 在相异节点 $z_0,z_1,\cdots,z_{2n+1}$ 上的插值多项式，利用牛顿插值多项式(2.4.7)，有

$$\begin{aligned}N_{2n+1}(x)=&f(z_0)+f[z_0,z_1](x-z_0)+f[z_0,z_1,z_2]\times\\&(x-z_0)(x-z_1)+\cdots+f[z_0,z_1,\cdots,z_{2n},z_{2n+1}]\times\\&(x-z_0)(x-z_1)\cdots(x-z_{2n+1}),\end{aligned} \qquad (2.6.14)$$

相应的余项表达式为

$$R_{2n+1}(x)=f[x,z_0,z_1,\cdots,z_{2n+1}](x-z_0)(x-z_1)\cdots(x-z_{2n+1}). \qquad (2.6.15)$$

在式(2.6.14)中应用重节点均差，特别设

$$z_{2i}=z_{2i+1}=x_i\quad(i=0,1,\cdots,n),$$

由式(2.6.14)得

$$N_{2n+1}(x)=f(x_0)+f[x_0,x_0](x-x_0)+f[x_0,x_0,x_1]\times$$
$$(x-x_0)^2+\cdots+f[x_0,x_0,\cdots,x_n,x_n]\times$$
$$(x-x_0)^2\cdots(x-x_{n-1})^2(x-x_n) \qquad (2.6.16)$$

该式是一个次数小于等于 $2n+1$ 的多项式. 它的误差可由 $z_{2i}\to x_i$, $z_{2i+1}\to x_i$ 时对式(2.6.15)取极限得到:

$$R_{2n+1}(x)=f[x,x_0,x_0,\cdots,x_n,x_n]\times$$
$$(x-x_0)^2(x-x_1)^2\cdots(x-x_n)^2. \qquad (2.6.17)$$

若 $f\in C^{2n+2}[a,b]$,则可得到与式(2.6.10)相同的表达式:

$$R_{2n+1}(x)=f(x)-N_{2n+1}(x)$$
$$=\frac{f^{(2n+2)}(\xi)}{(2n+2)!}W_{n+1}^2(x), \qquad (2.6.18)$$

其中 $\xi\in(a,b)$ 且依赖 x, $W_{n+1}(x)$ 仍由(2.2.7)定义.

由此可知式(2.6.16)得到的

$$N_{2n+1}(x)=\mathrm{H}_{2n+1}(x)$$

就是均差形式的埃尔米特插值多项式.

例 2.6.1 求多项式 p,满足条件

$$p(x_j)=f(x_j)\quad (j=0,1,2),$$
$$p'(x_1)=f'(x_1),$$

并求余项表达式.

解 为了保证满足 $p(x_j)=f(x_j)(j=0,1,2)$,设

$$p(x)=f(x_0)+f[x_0,x_1](x-x_0)+$$
$$f[x_0,x_1,x_2](x-x_0)(x-x_1)+$$
$$A(x-x_0)(x-x_1)(x-x_2).$$

这个等式右端前三项是以 x_0,x_1,x_2 为节点的牛顿均差插值公式,

从表中看出，随着 n 的增加 $R(x_{n-\frac{1}{2}})$ 的绝对值几乎成倍增加. 这说明当 $n\to\infty$ 时 $L_n(x)$ 在 $[-5,5]$ 上不收敛. 龙格证明了存在一个常数 $C\approx3.63$，使得当 $|x|\leqslant C$ 时，$\lim\limits_{n\to\infty}L_n(x)=f(x)$，而当 $|x|>C$ 时 $\{L_n(x)\}$ 发散.

下面取 $n=10$，根据计算画出 $y=L_{10}(x)$ 及 $y=\dfrac{1}{1+x^2}$ 在 $[-5,5]$ 上的图形，见图 2.3.

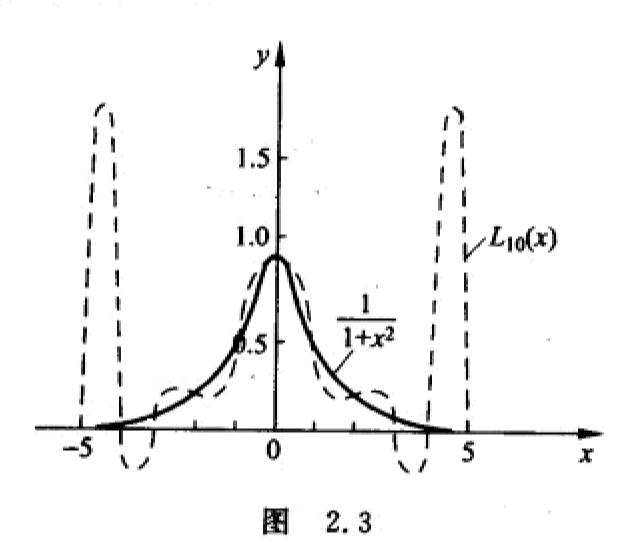


图 2.3

从图 2.3 看到，在 $x=\pm5$ 附近 $L_{10}(x)$ 与 $f(x)=1/(1+x^2)$ 偏离很远，例如 $L_{10}(4.8)=1.80438$，$f(4.8)=0.04160$. 这说明用高次插值多项式 $L_n(x)$ 近似 $f(x)$ 效果并不好，因而通常不用高次插值，而用分段低次插值. 从本例看到，如果把 $y=1/(1+x^2)$ 在节点 $x=0,\pm1,\pm2,\pm3,\pm4,\pm5$ 处用折线连起来显然比 $L_{10}(x)$ 逼近 $f(x)$ 好得多. 这正是 2.8 节要介绍的分段低次插值的出发点.

为得到 $\{L_n(x)\}$ 在 $[a,b]$ 上一致收敛于 $f(x)$ 的结论，先给出以下定理.

定理 2.7.3 设 $f(x)$ 在区间 $[a,b]$ 上无限次可导，则 $f(x)$ 为整函数的充分必要条件是

$$\lim_{n\to\infty}\frac{\sqrt[n]{M_n}}{n}=0,\quad 其中\quad M_n=\max_{a\leqslant x\leqslant b}|f^{(n)}(x)|.$$

定理 2.7.4 设 $f(x)$是$[a,b]$上的整函数，则对$[a,b]$上任给的节点阵(2.7.1)所构造的拉格朗日插值多项式序列$\{L_n(x)\}$一致收敛于 $f(x)$，即在$[a,b]$上

$$\lim_{n\to\infty} L_n(x) = f(x)$$

一致成立.

另一方面，对 $f(x)\in C[a,b]$，根据最佳一致逼近理论(见 3.6.2 节)可知，总存在一个节点阵使得由此构造的插值多项式序列$\{L_n(x)\}$收敛于 $f(x)$.

2.7.2 插值函数的稳定性

以上分析表明作函数的插值多项式时并非节点取得越多就能逼近得越好，此外，节点越多相应的高次插值多项式会发生激烈振荡，特别是在构造插值多项式时用到的节点函数值 $f(x_i)(i=0,1,\cdots,n)$，往往带有误差，因此随着插值多项式的不良性态，这种误差可能越来越扩大，那么函数值的误差经过插值过程将给计算带来什么影响呢？这就是插值函数的稳定性问题. 设 $f(x_i)$ $(i=0,1,\cdots,n)$有误差 δ_i，即实际构造插值函数时的函数表示为$(x_i,\tilde{f}_i)(i=0,1,\cdots,n)$，而精确值 $f(x_i)=\tilde{f}_i+\delta_i$，要研究插值函数 $I_n(f;x)$的误差是否增大，这里

$$I_n(f;x) = \sum_{i=0}^{n} l_i(x) f(x_i), \tag{2.7.3}$$

其中 $l_i(x)$是关于节点 $x_0,x_1,\cdots,x_n$ 的插值基函数，满足 $l_i(x_j)=\begin{cases}0, i\neq j,\\ 1, i=j\end{cases}$，当函数表示为$(x_i,\tilde{f}_i)(i=0,1,\cdots,n)$时，插值函数为

$$I_n(\tilde{f};x) = \sum_{i=0}^{n} l_i(x)\tilde{f}_i.$$

于是实际计算得到插值函数对于 $f(x)$的总误差为

余项为

$$R(x)=f(x)-p(x)=\frac{1}{4!}f^{(4)}(\xi)(x-x_0)^3(x-x_1),$$

其中 ξ 在 x_0,x_1,x 之间.

此例也可应用泰勒(Taylor)公式,设

$$p(x)=f(x_0)+f'(x_0)(x-x_0)+\frac{f''(x_0)}{2}(x-x_0)^2+A(x-x_0)^3.$$

显然,多项式 $p(x)$ 满足条件 $p^{(k)}(x_0)=f^{(k)}(x_0)(k=0,1,2)$,再由条件 $p(x_1)=f(x_1)$,可确定待定参数 A 为

$$A=\frac{1}{(x_1-x_0)^2}[f(x_0,x_1)-f'(x_0)]-\frac{f''(x_0)}{2(x_1-x_0)}.$$

2.7 插值多项式的收敛性与稳定性

2.7.1 插值多项式的收敛性与病态性质

插值多项式的收敛性问题是指在区间 $[a,b]$ 上的插值多项式 $p_n(x)$,当 n 增加时 $p_n(x)$ 是否收敛于被插函数 $f(x)$.

定义 2.7.1 设被插函数 $f(x)$ 对插值区间 $[a,b]$ 上的任何一个节点阵

$$\begin{cases} x_0^{(0)} \\ x_0^{(1)}, & x_1^{(1)} \\ x_0^{(2)}, & x_1^{(2)}, & x_2^{(2)} \\ \vdots & \vdots & \vdots & \ddots \\ x_0^{(n)}, & x_1^{(n)}, & x_2^{(n)}, & \cdots, & x_n^{(n)} \\ \vdots & \vdots & \vdots & & & \ddots \end{cases} \tag{2.7.1}$$

A 是待定常数. 利用条件 $p'(x_1)=f'(x_1)$ 定出常数

$$A=\frac{f'(x_1)-f[x_0,x_1]-(x_1-x_0)f[x_0,x_1,x_2]}{(x_1-x_0)(x_1-x_2)}.$$

因余项 $R(x)=f(x)-p(x)$ 满足条件:

$$R(x_j)=0\quad(j=0,1,2),$$
$$R'(x_1)=0.$$

可设

$$R(x)=K(x)(x-x_0)(x-x_1)^2(x-x_2),$$

其中 K 是待定函数. 应用微分学中著名的罗尔(Rolle)定理可定出 K,最后有

$$R(x)=\frac{1}{4!}f^{(4)}(\xi)(x-x_0)(x-x_1)^2(x-x_2),$$

其中 ξ 在 x_0,x_1,x_2,x 之间.

例 2.6.2 求多项式 p,满足条件

$$p(x_j)=f(x_j)\quad(j=0,1),$$
$$p^{(k)}(x_0)=f^{(k)}(x_0)\quad(k=1,2),$$

并求余项表达式.

解 此题可利用重节点均差形式表达式,令

$$p(x)=f(x_0)+f[x_0,x_0](x-x_0)+$$
$$f[x_0,x_0,x_1](x-x_0)^2+A(x-x_0)^2(x-x_1).$$

显然它满足条件 $p(x_i)=f(x_i)(i=0,1)$ 及 $p'(x_0)=f'(x_0)$, A 为待定参数,可由条件 $p''(x_0)=f''(x_0)$ 确定,即

$$p''(x_0)=2f[x_0,x_0,x_1]-(x_0+2x_1)A=f''(x_0),$$

得

$$A=\frac{2f[x_0,x_0,x_1]-f''(x_0)}{x_0+2x_1}.$$

$$E_n(f;x)=f(x)-I_n(\tilde{f};x)$$
$$=[f(x)-I_n(f;x)]+[I_n(f;x)-I_n(\tilde{f};x)],$$
取绝对值得
$$\max_{a\leqslant x\leqslant b}|E_n(f;x)|\leqslant\max_{a\leqslant x\leqslant b}|f(x)-I_n(f;x)|+$$
$$\max_{a\leqslant x\leqslant b}|I_n(f;x)-I_n(\tilde{f};x)|.$$
上式右端第一项为截断误差界，第二项为插值函数的舍人误差界，记作 ε_n，即
$$\varepsilon_n=\max_{a\leqslant x\leqslant b}|I_n(f;x)-I_n(\tilde{f};x)|$$
$$\leqslant\max_{a\leqslant x\leqslant b}\Big[\sum_{i=0}^{n}|l_i(x)|\Big]\max_{a\leqslant x\leqslant b}|f(x_i)-\tilde{f}_i|$$
$$=\lambda_n\max_{a\leqslant x\leqslant b}|f(x_i)-\tilde{f}_i|,\tag{2.7.4}$$
其中 $\lambda_n=\max\limits_{a\leqslant x\leqslant b}\left(\sum\limits_{i=0}^{n}|l_i(x)|\right)$. 若 λ_n 有界，则舍人误差也有界，且插值函数是稳定的. 下面先给出定义.

定义 2.7.5 对任给 $\varepsilon>0$，若存在 $\delta>0$，使当 $\max\limits_{a\leqslant x\leqslant b}|f(x_i)-\tilde{f}_i|\leqslant\delta$ 时，就有
$$\varepsilon_n=\max_{a\leqslant x\leqslant b}|I_n(f;x)-I_n(\tilde{f};x)|\leqslant\varepsilon,$$
则称插值函数 $I_n(f;x)$ 是**稳定的**，否则是**不稳定的**.

根据定义可得下面定理.

定理 2.7.6 若对一切 n，式(2.7.4)中的 λ_n 有界，则 n 次插值函数 $I_n(f;x)$ 是稳定的.

证明 对一切 n，λ_n 有界即 $\lambda_n<\lambda$，由定义 2.7.5，对任给的 $\varepsilon>0$，只要取 $\delta\leqslant\frac{\varepsilon}{\lambda}$，则由式(2.7.4)得
$$\varepsilon_n\leqslant\lambda_n\delta\leqslant\varepsilon,$$
于是 n 阶插值函数是稳定的.

其中 $x_i^{(k)} \neq x_j^{(k)}$ $(i \neq j, k=0,1,\cdots)$，相应的插值多项式序列$\{p_n(x)\}$满足$\lim\limits_{n\to\infty} p_n(x)=f(x)$，则称插值多项式 $p_n(x)$收敛于被插函数 $f(x)$.

对于 $f(x)$在$[a,b]$上按点阵(2.7.1)得到的拉格朗日插值多项式序列$\{L_n(x)\}$. 即使 $f(x)$在$[a,b]$上是无限次可导的函数也不能保证$\{L_n(x)\}$收敛于 $f(x)$，一个有名的实例是龙格(Runge)提供的.

例 2.7.2 $f(x)=\frac{1}{1+x^2}$在区间$[-5,5]$上各阶导数均存在，对给定的 n，将$[-5,5]$ n 等分，等距节点 $x_k=-5+\frac{10k}{n}$ $(k=0,1,\cdots,n)$，构造一个拉格朗日插值多项式为

$$L_n(x)=\sum_{i=0}^{n}\frac{1}{1+x_i^2}\frac{w_{n+1}(x)}{(x-x_i)w'_{n+1}(x_i)}. \qquad (2.7.2)$$

令 $x_{n-\frac{1}{2}}=\frac{1}{2}(x_{n-1}+x_n)$，则 $x_{n-\frac{1}{2}}=5-\frac{5}{n}$，表 2.5 列出了 $n=2,4,\cdots,20$ 的 $L_n(x_{n-\frac{1}{2}})$的计算结果及在 $x_{n-\frac{1}{2}}$的误差 $R(x_{n-\frac{1}{2}})$.

表 2.5

n	$f(x_{n-\frac{1}{2}})$	$L_n(x_{n-\frac{1}{2}})$	$R(x_{n-\frac{1}{2}})$
2	0.137931	0.759615	−0.621684
4	0.066390	−0.356826	0.423216
6	0.054463	0.607879	−0.553416
8	0.049651	−0.831017	0.880668
10	0.047059	1.578721	−1.531662
12	0.045440	−2.755000	2.800440
14	0.044334	5.332743	−5.288409
16	0.043530	−10.173867	10.217397
18	0.042920	20.123671	−20.080751
20	0.042440	−39.952449	39.994889

定理表明，插值函数 $I_n(f;x)$ 是否稳定取决于插值基函数 $l_i(x)(i=0,1,\cdots,n)$ 的性质，与被插值函数 $f(x)$ 无关，对拉格朗日插值多项式

$$L_n(x)=\sum_{i=0}^{n}\frac{W_{n+1}(x)}{(x-x_i)W'_{n+1}(x_i)}f(x_i),$$

此时 $\lambda_n=\max\limits_{a\leqslant x\leqslant b}\sum\limits_{i=0}^{n}|l_i(x)|=\max\limits_{a\leqslant x\leqslant b}\sum\limits_{i=0}^{n}\left|\frac{W_{n+1}(x)}{(x-x_i)W'_{n+1}(x_i)}\right|.$

可以证明不管节点阵(2.7.1)如何取，都有 $\lambda_n>\frac{1}{8\sqrt{\pi}}\ln n$，所以 λ_n 无界. 因此，拉格朗日插值多项式总不稳定，而且其舍入误差增长速度至少与 $O(\ln n)$ 同阶. 特别当插值节点等距分布时，还可证明 $\lambda_n\approx\frac{\sqrt{2}}{\pi}\frac{2^n}{n^2}$，说明 λ_n 增长很快.

从上面插值函数收敛性与稳定性分析都能看到高次插值多项式 $L_n(x)$ 作为 $f(x)$ 的逼近是不合适的，应尽量避免，为此实际应用多采用分段低次插值.

2.8 分段低次插值

2.8.1 分段线性插值

分段线性插值是用折线逼近函数，是最简单的分段低次插值.

用 f 表示区间 $[a,b]$ 上的函数，引进记号

$$C^k[a,b]=\{f\mid f^{(k)}\text{ 在}[a,b]\text{ 上连续}\},$$

即 $C^k[a,b]$ 是 $[a,b]$ 上具有连续 k 阶导数的函数全体所组成的集合. 特别地，记

$$C[a,b]=C^0[a,b],$$

这是 $[a,b]$ 上连续函数全体所组成的集合. 以后可以用记号 $f\in C^k[a,b]$ 表示 f 是 $[a,b]$ 具有连续 k 阶导数的函数.

定义 2.8.1 设 f 是$[a,b]$上的函数，在节点 $a=x_0<x_1<\cdots<x_n=b$ 上的函数值为 $f_0,f_1,\cdots,f_n$. 记 $h=\max\limits_{0\leqslant i\leqslant n-1} h_i$; $h_i=x_{i+1}-x_i$ $(i=0,1,\cdots,n-1)$. 如果函数 I_h 满足下列条件，则称 I_h 是 f 的**分段线性插值函数**.

(1) $I_h\in C[a,b]$;

(2) $I_h(x_i)=f_i(i=0,1,\cdots,n)$;

(3) 在每个子区间$[x_i,x_{i+1}](0\leqslant i\leqslant n-1)$上，$I_h$ 是次数不超过 1 的多项式.

依据定义，I_h 在子区间$[x_i,x_{i+1}]$上有

$$I_h(x)=f_i\frac{x-x_{i+1}}{x_i-x_{i+1}}+f_{i+1}\frac{x-x_i}{x_{i+1}-x_i}\quad(x_i\leqslant x\leqslant x_{i+1}). \tag{2.8.1}$$

由此可得 I_h 在整个区间$[a,b]$上的表达式

$$I_h(x)=\sum_{i=0}^{n}f_il_i(x), \tag{2.8.2}$$

其中 l_i 是分段线性插值的基函数. l_i 的定义是

$$l_i(x)=\begin{cases}0, & x\notin[x_{i-1},x_{i+1}],\\ \dfrac{x-x_{i-1}}{x_i-x_{i-1}}, & x\in[x_{i-1},x_i](i=0\text{ 时略去}),\\ \dfrac{x-x_{i+1}}{x_i-x_{i+1}}, & x\in[x_i,x_{i+1}](i=0\text{ 时略去}).\end{cases} \tag{2.8.3}$$

下面的定理表明，分段线性插值能任意逼近连续函数.

定理 2.8.2 设 $f\in C[a,b]$，则当 $h\to0$ 时 I_h 一致收敛于 f，即对任意一常数 $\varepsilon>0$，均存在相应的 $\delta>0$，只要 $h<\delta$，便有$|f(x)-I_h(x)|<\varepsilon$, $\forall x\in[a,b]$.

如果 f 在$[a,b]$上存在二阶导数，利用线性插值余项的界(2.2.12)，可得分段线性插值的余项 $R(x)=f(x)-I_h(x)$的一个界

$$|R(x)| \leqslant \frac{Mh^2}{8}, \quad M = \max_{a \leqslant x \leqslant b} |f''(x)|. \qquad (2.8.4)$$

这时定理 2.8.2 的结论是明显的.

2.8.2 分段三次埃尔米特插值

分段线性插值函数在节点处导数不存在.如果需要插值函数在节点处也可导,并且在节点处除了提供函数值外还提供了导数值,则可进行分段埃尔米特插值.

定义 2.8.3 设函数 f 在节点 $a=x_0<x_1<\cdots<x_n=b$ 上的函数为 $f_0,f_1,\cdots,f_n$,导数值为 $f'_0, f'_1, \cdots, f'_n$. 如果函数 I_h 满足下列条件,则称 I_h 是 f 的**分段埃尔米特插值函数**.

(1) $I_h \in C^1[a,b]$,

(2) $I_h(x_i)=f_i, I'_h(x_i)=f'_i \ (i=0,1,\cdots,n)$,

(3) 在每个子区间$[x_i,x_{i+1}]$$(0\leqslant i\leqslant n-1)$上,$I_h$ 是次数不超过 3 的多项式,

由两点三次插值公式(2.6.11),I_h 在子区间$[x_i,x_{i+1}]$上有

$$\begin{aligned} I_h(x) = & f_i\left(1+2\frac{x-x_i}{x_{i+1}-x_i}\right)\left(\frac{x-x_{i+1}}{x_i-x_{i+1}}\right)^2 + \\ & f_{i+1}\left(1+2\frac{x-x_{i+1}}{x_i-x_{i+1}}\right)\left(\frac{x-x_i}{x_{i+1}-x_i}\right)^2 + \\ & f'_i(x-x_i)\left(\frac{x-x_{i+1}}{x_i-x_{i+1}}\right)^2 + \\ & f'_{i+1}(x-x_{i+1})\left(\frac{x-x_i}{x_{i+1}-x_i}\right)^2, \end{aligned} \qquad (2.8.5)$$

由此可得 I_h 在整个区间$[a,b]$上的表达式

$$I_h(x) = \sum_{i=0}^{n}[f_i\alpha_i(x) + f'_i \ \beta_i(x)], \qquad (2.8.6)$$

其中 α_i,β_i 是分段三次埃尔米持插值的基函数. α_i,β_i 的定义是

$$\alpha_i(x)=\begin{cases}0, & x\notin[x_{i-1},x_{i+1}],\\ \left(1+2\dfrac{x-x_i}{x_{i-1}-x_i}\right)\left(\dfrac{x-x_{i-1}}{x_i-x_{i-1}}\right)^2, & x\in[x_{i-1},x_i]\ (i=0\text{ 时略去}),\\ \left(1+2\dfrac{x-x_i}{x_{i+1}-x_i}\right)\left(\dfrac{x-x_{i+1}}{x_i-x_{i+1}}\right)^2, & x\in[x_i,x_{i+1}]\ (i=n\text{ 时略去}),\end{cases}\tag{2.8.7}$$

$$\beta_i(x)=\begin{cases}0, & x\notin[x_{i-1},x_{i+1}],\\ (x-x_i)\left(\dfrac{x-x_{i-1}}{x_i-x_{i-1}}\right)^2, & x\in[x_{i-1},x_i]\ (i=0\text{ 时略去}),\\ (x-x_i)\left(\dfrac{x-x_{i+1}}{x_i-x_{i+1}}\right)^2, & x\in[x_i,x_{i+1}]\ (i=n\text{ 时略去}).\end{cases}\tag{2.8.8}$$

可以证明，若 $f\in C[a,b]$，$h=\max\limits_{0\leqslant i\leqslant n-1}(x_{i+1}-x_i)$，则当 $h\to 0$ 时，分段三次埃尔米特插值函数 I_h 一致收敛于 f.

2.9 样条插值

2.9.1 样条函数

样条插值也是用分段低次多项式去逼近函数. 在分段低次插值中，分段线性插值函数在节点处不可导，分段三次埃尔米特插值函数有连续一阶导数，光滑性也较差，而且需要提供各个节点处的导数值. 样条插值能较好地适应对光滑性的不同需求，并且只需在插值区间端点提供某些导数信息.

样条插值是用样条函数去逼近函数.

定义 2.9.1 设在区间$[a,b]$上给定一个分划

$$a=x_0<x_1<\cdots<x_n=b,$$

如果函数 S 满足条件，则称 S 是以 $x_0,x_1,\cdots,x_n$ 为节点的 ***k* 次样条函数**(spline of degree k).

(1) $S\in C^{k-1}[a,b]$；

(2) 在每个子区间$[x_i,x_{i+1}]$$(0\leqslant i\leqslant n-1)$上，$S$ 是次数不超过k 的多项式.

近几十年来，函数逼近在理论研究和实际应用中均获得重大的进展，其中非常流行和十分活跃的分支是样条函数. 在飞机、船舶、汽车等外形设计的工程问题中，在数值微分、数值积分、微分方程和积分方程的数值解法，以及观测和实验数据的处理等方面，样条函数都有着重要的应用.

2.9.2 *B* 样条

B 样条是一类基本的样条函数.

定义 2.9.2 记

$$u_+^m=\begin{cases}u^m, & \text{当 } u\geqslant 0,\\ 0, & \text{当 } u<0,\end{cases}\quad (m=1,2,\cdots),$$

称之为 **m 次半截单项式**. 并规定

$$u_+^0=\begin{cases}1, & \text{当 } u>0,\\ \dfrac{1}{2}, & \text{当 } u=0,\\ 0, & \text{当 } u<0.\end{cases}$$

利用定义 2.9.2 可以直接给出 B 样条的定义.

定义 2.9.3 设 Ω_k 是$(-\infty,\infty)$上的函数，

$$\Omega_k(x)=\frac{1}{k!}\sum_{j=0}^{k+1}(-1)^j\binom{k+1}{j}\left(x+\frac{k+1}{2}-j\right)_+^k,$$

称之为 **k 次 B 样条或基本样条函数**.

可以看出，Ω_k 是一个分段 k 次多项式，k 阶导数间断点为

$$x_j=j-\frac{k+1}{2}\quad (j=0,1,\cdots,k+1),$$

因此，Ω_k 是以 $x_0, x_1, \cdots, x_{k+1}$ 为节点的 k 次样条函数. Ω_k 在整个实数轴$(-\infty,\infty)$上有定义，但是容易证明，它在区间$[x_0, x_{k+1}]$，即$\left(-\frac{k+1}{2},\frac{k+1}{2}\right)$之外恒为零. 下面是几个低次 B 样条的表达式及相应的图形.

0～3 次 B 样条表达式：

$$\Omega_0(x)=\begin{cases}0, & |x|>\frac{1}{2},\\ 1, & |x|<\frac{1}{2},\\ \frac{1}{2}, & |x|=\frac{1}{2};\end{cases} \tag{2.9.1}$$

$$\Omega_1(x)=\begin{cases}0, & |x|\geqslant 1,\\ 1-|x|, & |x|<1;\end{cases}$$

$$\Omega_2(x)=\begin{cases}0, & |x|>\frac{3}{2},\\ -x^2+\frac{3}{4}, & |x|<\frac{1}{2},\\ \frac{1}{2}x^2-\frac{3}{2}|x|+\frac{9}{8}, & \frac{1}{2}\leqslant|x|\leqslant\frac{3}{2};\end{cases}$$

$$\Omega_3(x)=\begin{cases}0, & |x|\geqslant 2,\\ \frac{1}{2}|x|^3-x^2+\frac{2}{3}, & |x|\leqslant 1,\\ -\frac{1}{6}|x|^3+x^2-2|x|+\frac{4}{3}, & 1<|x|<2.\end{cases}$$

从图 2.4 可以看出，随次数的增加，B 样条的曲线越来越光滑，节点(图中的圆点)越来越多. 实际上，定义 2.9.3 中 Ω_k 的表达式是由下述递推公式得出的：

$$\Omega_i(x)=\int_{x-\frac{1}{2}}^{x+\frac{1}{2}}\Omega_{i-1}(x)\mathrm{d}x \quad (i=1,2,\cdots), \tag{2.9.2}$$

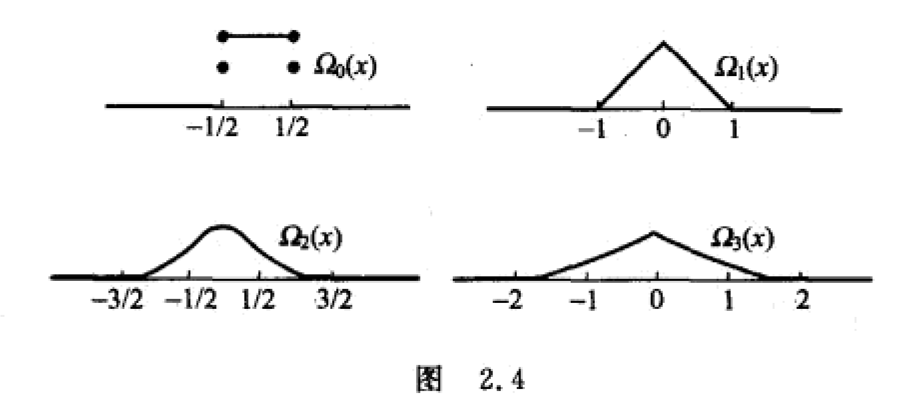


图 2.4

其中 Ω_0 已在式(2.9.1)中给出.由于积分运算可以使间断函数变得连续,使粗糙的函数曲线变得光滑,因此,次数越高的 B 样条其曲线越光滑.

在样条函数的研究中,无论是理论分析还是实际数值计算,B 样条 Ω_k 都有它的独特作用.

2.9.3 三次样条插值问题的提法

三次样条是应用最广泛的样条,它有明确的力学意义.早期绘图员在制图时,用一种富有弹性的细长木条,称为样条,把它用压铁固定在若干样点上后画出的曲线称为样条曲线.这种样条曲线在数学上表现为符合定义 2.9.1 的三次样条函数.

定义 2.9.4 设在区间$[a,b]$上给定一个分划

$$a = x_0 < x_1 < \cdots < x_n = b,$$

S 是以 $x_0, x_1, \cdots, x_n$ 为节点的三次样条函数.如果 S 满足插值条件

$$S(x_i) = y_i \quad (i = 0, 1, \cdots, n), \tag{2.9.3}$$

则称 S 为**三次样条插值函数**.

由定义,关于 S 的可供利用的条件有 $4n-2$ 个,其中插值条件包含了 $n+1$ 个;另由 $S \in C^2[a,b]$,节点处的连续性条件有 $3n-3$ 个:

2.9.4 均匀分划的三次样条插值函数

设分划是均匀的,即

$$x_i = a + ih \quad (i = 0,1,\cdots,n), \quad h = \frac{b-a}{n},$$

这时,有一组三次 B 样条

$$\Omega_3\left(\frac{x-x_i}{h}\right) = \Omega_3\left(\frac{x-x_0}{h} - i\right), \tag{2.9.8}$$

$$i = -1,0,1,\cdots,n+1,$$

它们中的每一个在$[a,b]$上都不恒等于零,其他 i 所对应的 B 样条在$[a,b]$则恒为零(见图 2.5).

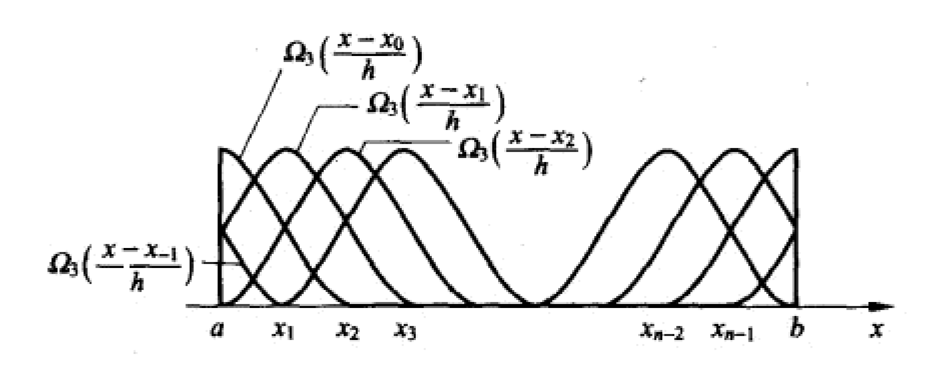


图 2.5

三次 B 样条(2.9.8)中共有 $n+3$ 个 B 样条,考虑以它们的线性组合作为插值函数 S,

$$S(x) = \sum_{j=-1}^{n+1} c_j \Omega_3\left(\frac{x-x_0}{h} - j\right), \tag{2.9.9}$$

其中 $c_j(-1 \leqslant j \leqslant n+1)$是待定常数. 这样定义的 S 明显是以 x_0, $x_1,\cdots,x_n$ 为节点的三次样条函数,式(2.9.4)中的 $3n-3$ 个条件能自动地满足. 因此剩下的插值条件(2.9.3)及附加边界条件共包含 $n+3$ 个条件恰好可确定 $c_{-1},c_0,\cdots,c_{n+1}$.

对问题 I 根据条件(2.9.3)及(2.9.5),有

$$\begin{cases} S(x_i-0)=S(x_i+0), \\ S'(x_i-0)=S'(x_i+0), \\ S''(x_i-0)=S''(x_i+0), \end{cases} \tag{2.9.4}$$

$i=1,2,\cdots,n-1$. 但是,S 由 n 段次数不超过 3 的多项式组成,共有$4n$个待定参数. 因此,为了确定 S,还缺少两个条件. 通常是在区间$[a,b]$的端点 $a=x_0$,$b=x_n$ 上各附加一个条件. 在端点上的条件称为**边界条件**. 插值条件(2.9.3)中已包含两个边界条件. 常见的附加边界条件有三种,并由此提出以下三个类型的插值问题.

问题Ⅰ 求三次样条插值函数 S,满足附加边界条件

$$S'(x_0)=y_0',\quad S'(x_n)=y_n'. \tag{2.9.5}$$

问题Ⅱ 求三次样条插值函数 S,满足附加边界条件

$$S''(x_0)=y_0'',\quad S''(x_n)=y_n''. \tag{2.9.6}$$

其特殊情况

$$S''(x_0)=S''(x_n)=0,$$

称为**自然边界条件**.

问题Ⅲ 当被插函数是以 x_n-x_0 为周期的周期函数时,要寻求三次样条 S,满足内部节点处的条件

$$S(x_i)=y_i\quad(i=1,2,\cdots,n-1)$$

及边界条件

$$\begin{aligned} &S(x_0)=S(x_n)=y_0, \\ &S^{(k)}(x_0+0)=S^{(k)}(x_n-0)\quad(k=1,2). \end{aligned} \tag{2.9.7}$$

这是周期型问题,所寻求的样条函数 S 称为**周期样条函数**(periodic spline function).

定理 2.9.5 插值问题Ⅰ、Ⅱ、Ⅲ的解均存在惟一. 因此,问题是如何具体去构造三次样条插值函数 S.

$$\boldsymbol{f} = \begin{pmatrix} 6y_0 + 2hy_0' \\ 6y_1 \\ 6y_2 \\ \vdots \\ 6y_{n-1} \\ 6y_n - 2hy_n' \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{n-1} \\ c_n \end{pmatrix}.$$

还有关系式

$$\begin{cases} c_{-1} = c_1 - 2hy_0', \\ c_{n+1} = c_{n-1} + 2hy_n'. \end{cases} \tag{2.9.13}$$

对以三对角矩阵为系数矩阵的线性代数方程组有方便的解法(见第 6 章).从方程(2.9.12)解出 $c_0, c_1, \cdots, c_n$,再从式(2.9.13)算出 c_{-1}, c_{n+1},然后代入式(2.9.9),便得所求样条插值函数 S.

对问题Ⅱ 由条件(2.9.3)及(2.9.6),可归结出如下代数方程组:

$$\boldsymbol{a}\boldsymbol{c} = \boldsymbol{f}, \tag{2.9.14}$$

其中

$$\boldsymbol{A} = \begin{pmatrix} 4 & 1 & & & & \\ 1 & 4 & 1 & & & \\ & 1 & 4 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & 4 & 1 \\ & & & & 1 & 4 \end{pmatrix}$$

是 $n-1$ 阶严格对角占优的对称三对角矩阵,而

$$\boldsymbol{f} = \begin{pmatrix} 6y_1 - y_0 + \frac{h^2}{6}y_0'' \\ 6y_2 \\ 6y_3 \\ \vdots \\ 6y_{n-2} \\ 6y_{n-1} - y_n + \frac{h^2}{6}y_n'' \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_{n-2} \\ c_{n-1} \end{pmatrix}.$$

还有关系式

$$
\begin{cases}
c_{-1} = 2c_0 - c_1 + h^2 y_0'' , \\
c_0 = y_0 - \dfrac{h^2}{6} y_0'' , \\
c_n = y_n - \dfrac{h^2}{6} y_n'' , \\
c_{n+1} = 2c_n - c_{n-1} + h^2 y_n'' .
\end{cases}
\tag{2.9.15}
$$

从方程(2.9.14)解出 $c_1, c_2, \cdots, c_{n-1}$，再从式(2.9.15)中算出 c_{-1}，c_0, c_n, c_{n+1}.

对问题Ⅲ 由条件(2.9.3)及(2.9.7)可以推出

$$
c_{n+1} = c_1, \quad c_0 = c_n, \quad c_{-1} = c_{n-1} \tag{2.9.16}
$$

及线性代数方程组

$$
\boldsymbol{ac} = \boldsymbol{f}, \tag{2.9.17}
$$

其中

$$
\boldsymbol{A} = \begin{pmatrix}
4 & 1 & & & & \\
1 & 4 & 1 & & & \\
 & 1 & 4 & 1 & & \\
 & & \ddots & \ddots & \ddots & \\
 & & & 1 & 4 & 1 \\
 & & & & 1 & 4
\end{pmatrix}
$$

是 n 阶矩阵，而

$$
\boldsymbol{f} = \begin{pmatrix} 6y_1 \\ 6y_2 \\ 6y_3 \\ \vdots \\ 6y_{n-1} \\ 6y_0 \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_{n-1} \\ c_n \end{pmatrix}.
$$

$$
\begin{cases}
S'(x_0) = \dfrac{1}{h}\sum_{j=-1}^{n+1} c_j \Omega_3'(-j) = y_0', \\
S(x_i) = \sum_{j=-1}^{n+1} c_j \Omega_3(i-j) = y_i, \\
S'(x_n) = \dfrac{1}{h}\sum_{j=-1}^{n+1} c_j \Omega_3'(n-j) = y_n'
\end{cases}
\tag{2.9.10}
$$

$i=0,1,\cdots,n$. 利用式(2.9.1)中 Ω_3 的表达式,可得

$$\Omega_3(0)=\frac{2}{3},\quad \Omega_3(\pm 1)=\frac{1}{6},\quad \Omega_3(x)=0 \quad (|x|\geqslant 2).$$

$$\Omega_3'(0)=0,\quad \Omega_3'(\pm 1)=\mp\frac{1}{2},\quad \Omega_3'(x)=0 \quad (|x|\geqslant 2).$$

于是式(2.9.10)的具体形式为

$$
\begin{cases}
\dfrac{-c_{-1}+c_1}{2h} = y_0', \\
\dfrac{1}{6}c_{i-1}+\dfrac{2}{3}c_i+\dfrac{1}{6}c_{i+1} = y_i, \quad i=0,1,\cdots,n. \\
\dfrac{-c_{n-1}+c_{n+1}}{2h} = y_n',
\end{cases}
\tag{2.9.11}
$$

最后归结为解线性代数方程组:

$$\boldsymbol{ac}=\boldsymbol{f}, \tag{2.9.12}$$

其中

$$
\boldsymbol{A}=\begin{pmatrix}
4 & 2 & & & & \\
1 & 4 & 1 & & & \\
 & 1 & 4 & 1 & & \\
 & & \ddots & \ddots & \ddots & \\
 & & & 1 & 4 & 1 \\
 & & & & 2 & 4
\end{pmatrix}
$$

是 $n+1$ 阶具有严格对角优势的三对角矩阵,而

从式(2.9.17)解出 $c_1, c_2, \cdots, c_n$，由式(2.9.16)可直接确定 c_{n+1}, c_0, c_{-1}.

2.9.5 任意分划的三次样条插值函数

假设分划是任意的，构造三次样条插值函数 S 的一种方法如下. 令

$$S''(x_i) = M_i \quad (i = 0, 1, \cdots, n), \tag{2.9.18}$$

由于 S 是分段次数不超过 3 的多项式，而且二阶导数 S'' 连续，可知 S'' 是折线函数，因此在任意取定的子区间 $[x_i, x_{i+1}]$ 上有

$$S''(x) = M_i \frac{x_{i+1} - x}{h_i} + M_{i+1} \frac{x - x_i}{h_i}, \tag{2.9.19}$$

其中 $h_i = x_{i+1} - x_i$. 将式(2.9.19)积分两次，并利用插值条件

$$S(x_i) = y_i, \quad S(x_{i+1}) = y_{i+1},$$

确定两个积分常数，得到

$$\begin{aligned} S(x) = & M_i \frac{(x_{i+1} - x)^3}{6h_i} + M_{i+1} \frac{(x - x_i)^3}{6h_i} + \\ & \left(\frac{y_i}{h_i} - \frac{M_i h_i}{6}\right)(x_{i+1} - x) + \\ & \left(\frac{y_{i+1}}{h_i} - \frac{M_{i+1} h_i}{6}\right)(x - x_i) \quad (x \in [x_i, x_{i+1}]), \end{aligned} \tag{2.9.20}$$

对 S 求导，又得

$$\begin{aligned} S'(x) = & -M_i \frac{(x_{i+1} - x)^2}{2h_i} + M_{i+1} \frac{(x - x_i)^2}{2h_i} + \frac{y_{i+1} - y_i}{h_i} - \\ & \frac{M_{i+1} - M_i}{6} h_i \quad (x \in [x_i, x_{i+1}]). \end{aligned} \tag{2.9.21}$$

利用式(2.9.4)中的条件

$$S'(x_i - 0) = S'(x_i + 0) \quad (i = 1, 2, \cdots, n - 1),$$

并根据在子区间$[x_{i-1},x_i]$及$[x_i,x_{i+1}]$上S'的两种表达式，可得方程

$$\mu_i M_{i-1} + 2M_i + \lambda_i M_{i+1} = d_i \quad (i = 1,2,\cdots,n-1), \tag{2.9.22}$$

其中

$$\begin{cases} \mu_i = \dfrac{h_{i-1}}{h_{i-1} - h_i}, \\ \lambda_i = 1 - \mu_i, \\ d_i = \dfrac{6}{h_{i-1} + h_i}\left(\dfrac{y_{i+1} - y_i}{h_i} - \dfrac{y_i - y_{i-1}}{h_{i-1}}\right). \end{cases} \tag{2.9.23}$$

式(2.9.22)是关于$n+1$个未知数$M_0, M_1, \cdots, M_n$的$n-1$个方程，尚需附加边界条件，补充两个方程.

对问题Ⅰ 由条件(2.9.5)并利用式(2.9.21)可导出如下两个方程：

$$\begin{cases} 2M_0 + M_1 = \dfrac{6}{h_0}\left(\dfrac{y_1 - y_0}{h_0} - y'_0\right), \\ M_{n-1} + 2M_n = \dfrac{6}{h_{n-1}}\left(y'_n - \dfrac{y_n - y_{n-1}}{h_{n-1}}\right). \end{cases} \tag{2.9.24}$$

对问题Ⅱ 由条件(2.9.6)直接有

$$\begin{cases} M_0 = y''_0, \\ M_n = y''_n. \end{cases} \tag{2.9.25}$$

对问题Ⅲ 由条件(2.9.7)有

$$y_n = y_0,$$

而且可导出两个方程：

$$\begin{cases} M_n = M_0, \\ \lambda_n M_1 + \mu_n M_{n-1} + 2M_n = d_n, \end{cases} \tag{2.9.26}$$

其中

$$\begin{cases}\mu_n = \dfrac{h_{n-1}}{h_0 + h_{n-1}}, \\ \lambda_n = 1 - \mu_n, \\ d_n = \dfrac{6}{h_0 + h_{n-1}}\left(\dfrac{y_1 - y_0}{h_0} - \dfrac{y_n - y_{n-1}}{h_{n-1}}\right).\end{cases} \tag{2.9.27}$$

由式(2.9.22)与式(2.9.24),或式(2.9.22)与式(2.9.25),或式(2.9.22)与式(2.9.26),分别构成具有三对角非奇异系数矩阵的方程组,可以解出 $M_0, M_1, \cdots, M_n$,并由式(2.9.20)与式(2.9.21)确定样条插值函数及其导函数. 这样,插值问题Ⅰ、Ⅱ、Ⅲ均得到解决.

2.9.6 三次样条插值的收敛性

利用三次样条插值,不会出现拉格朗日插值时的那种"龙格现象". 为了提高插值的精确度,样条插值可用增加节点的办法来实现,而拉格朗日插值增加节点会导致多项式次数的增高,常引起计算的不稳定. 下面仍取第 2.7 节中龙格提供的函数为数值计算的实例.

例 2.9.6 设 $f(x)=\dfrac{1}{1+x^2}, x\in[-5,5]$,节点为

$$x_i = -5 + i \quad (i = 0,1,\cdots,10).$$

求三次样条插值函数 S,满足插值条件

$$S(x_i) = f(x_i) \quad (i = 0,1,\cdots,10),$$

及附加边界条件

$$S'(-5) = f'(-5), \quad S'(5) = f'(5),$$

即对函数 f 求样条插值问题Ⅰ的解.

解 表 2.6 给出了 S 的数值结果,为了比较,表中还列出了 f 及拉格朗日插值多项式 L_{10} 的相应结果. L_{10} 的几何曲线已在图 2.3 中给出. 可以看出,S 能很好地近似 f.

表 2.6

x	$\frac{1}{1+x^2}$	$S(x)$	$L_{10}(x)$
−5.0	0.03846	0.03846	0.03846
−4.8	0.04160	0.03758	1.80438
−4.3	0.05131	0.04842	0.88808
−4.0	0.05882	0.05882	0.05882
−3.8	0.06477	0.06556	−0.20130
−3.3	0.08410	0.08426	−0.10832
−3.0	0.10000	0.10000	0.10000
−2.8	0.11312	0.11366	0.19837
−2.3	0.15898	0.16115	0.24145
−2.0	0.20000	0.20000	0.20000
−1.8	0.23585	0.23154	0.18878
−1.3	0.37175	0.36133	0.31650
−1.0	0.50000	0.50000	0.50000
−0.8	0.60975	0.62420	0.64316
−0.3	0.91743	0.92754	0.94090
0.0	1.00000	1.00000	1.00000

对于 $f\in C[a,b]$，记

$$\|f\|_\infty=\max_{a\leqslant x\leqslant b}|f(x)|,$$

称 $\|f\|_\infty$ 为函数 f 的∞范数.

定理 2.9.7 给出了关于三次样条插值收敛性的一个结论.

定理 2.9.7 设 $f\in C^4[a,b]$，S 为问题Ⅰ或问题Ⅱ的插值函数，令

$$\beta=\frac{h}{\min\limits_{0\leqslant i\leqslant n-1}h_i},\quad h=\max_{0\leqslant i\leqslant n-1}h_i,\quad h_i=x_{i+1}-x_i$$

$$(i=0,1,\cdots,n-1),$$

β 称为分划比. 则有估计式

$$\|f^{(k)}-S^{(k)}\|_\infty\leqslant c_k\|f^{(4)}\|_\infty h^{4-k}\quad(k=0,1,2,3),\tag{2.9.28}$$

其中

$$c_0=\frac{5}{384},\quad c_1=\frac{1}{24},\quad c_2=\frac{3}{8},\quad c_3=\frac{\beta+\beta^{-1}}{2}.$$

这个定理中系数 c_0, c_1, c_2 均与分划比无关，仅 c_3 与分划比 β 有关. 因此，当 $h\to 0$ 时，式(2.9.28)表明，样条插值函数 S 及其一阶导数 S'，二阶导数 S''，分别一致收敛于 f, f', f''. 若分划比在加密过程中能保证

$$0<m\leqslant\beta\leqslant M,$$

这里 m 与 M 是常数，则还有 S''' 一致收敛于 f'''.

2.10 反插值

2.10.1 插值与反插值

设函数 f 的离散数据为

$$(x_i, y_i),\quad y_i=f(x_i),\quad i=0,1,\cdots,n,$$

插值的目的是在 $x_0, x_1, \cdots, x_n$ 之间给定了自变量 x 的值后，要去求函数 f 的近似值，其途径是构造插值多项式. 不同的构造方法，就是不同的插值法. 与此相反，反插值的目的是在 $y_0, y_1, \cdots, y_n$ 之间给定了函数 f 的值后，要去求自变量 x 的近似值，其途径仍是利用插值法.

反插值有两种处理方式：一种是直接利用函数 f 的插值多项式；另一种是在假设 f^{-1} 存在的前提下，构造 f 的反函数 f^{-1} 的插值多项式.

因为反插值归结为求满足 $f(x)=c$ 的 x 的近似值，这里 c 是在 $y_0, y_1, \cdots, y_n$ 之间的某个值. 如果 f 的反函数 f^{-1} 存在，则 $x=f^{-1}(c)$.

反插值就是求 f^{-1} 在 c 上的近似值. 当 $c=0$ 时，反插值就是求函数 f 的近似零点，或者说是求方程 $f(x)=0$ 的近似根. 所以反插值有明显的意义.

2.10.2 利用函数的插值多项式反插

考虑在 x_0 与 x_1 之间进行反插值. 假设 c 是 $y_0=f(x_0)$ 与 $y_1=f(x_1)$ 之间的一个值,则当 f 连续时,在 x_0 与 x_1 之间必存在 x,使得 $f(x)=c$,寻求 x 的近似值,一般只利用 f 的低次插值多项式.

从离散数据算出 f 的均差表,如果均差表中的二阶均差实际上可看作零,则 f 在 x_0 与 x_1 之间可用一次插值多项式 p_1 近似代替,

$$p_1(x)=f(x_0)+f[x_0,x_1](x-x_0). \qquad (2.10.1)$$

令 $p_1(x)=c$,从式(2.10.1)中解出 x,即得所要求的近似值.

如果均差表中到三阶均差才实际上可看为零,则 f 在 x_0 与 x_1 之间可用二次插值多项式 p_2 近似代替,

$$p_2(x)=f(x_0)+f[x_0,x_1](x-x_0)+f[x_0,x_1,x_2](x-x_0)(x-x_1). \qquad (2.10.2)$$

这时,令 $p_2(x)=c$,要从式(2.10.2)去解出 x,一般用逐次逼近法:先取 $x^{(0)}$,使得满足

$$f(x_0)+f[x_0,x_1](x^{(0)}-x_0)=c,$$

即

$$x^{(0)}=x_0+\frac{c-f(x_0)}{f[x_0,x_1]}.$$

再求 $x^{(1)}$,使得满足

$$f(x_0)+f[x_0,x_1](x^{(1)}-x_0)+f[x_0,x_1,x_2](x^{(0)}-x_0)(x^{(0)}-x_1)=c,$$

即

$$x^{(1)}=x_0+\frac{c-f(x_0)}{f[x_0,x_1]}-\frac{f[x_0,x_1,x_2]}{f[x_0,x_1]}(x^{(0)}-x_0)(x^{(0)}-x_1).$$

然后可利用迭代公式

$$x^{(k)}=x^{(0)}-\frac{f[x_0,x_1,x_2]}{f[x_0,x_1]}(x^{(k-1)}-x_0)(x^{(k-1)}-x_1) \tag{2.10.3}$$

反复进行迭代,直至 $x^{(k)}$ 与 $x^{(k-1)}$ 在所要求的精确度下相等为止.

如果在均差表中三阶均差实际上还不是零,则可用更高次的插值多项式.

对于等距节点情形,可改用差分形式的插值公式.设

$$x_i=x_0+ih \quad (i=0,1,\cdots,n),$$

且

$$x=x_0+th \quad (0<t<1),$$

则迭代公式(2.10.3)可改写成

$$t^{(k)}=t^{(0)}-\frac{\Delta^2 y_0}{2\Delta y_0}t^{(k-1)}(t^{(k-1)}-1) \quad (k=1,2,\cdots), \tag{2.10.4}$$

其中

$$t^{(0)}=\frac{c-y_0}{\Delta y_0}.$$

例 2.10.1 已知 $f(x)=\sin x$ 的函数表如下:

x_i	$y_i=\sin x_i$	x_i	$y_i=\sin x_i$
1.72	0.9888898	1.78	0.9781966
1.74	0.9857192	1.80	0.9738476
1.76	0.9821543	1.82	0.9691091

求 arcsin0.9800000 的值.

解 取 $x_0=1.76$, $y_0=0.9821543$. 经计算得

$$\Delta y_0=-0.0039577, \quad \Delta^2 y_0=-0.0003913,$$

$$t^{(0)}=\frac{0.9800000-0.9821543}{-0.0039577}=0.5443313.$$

采用二次插值多项式，将 $t^{(0)}$，Δy_0，$\Delta^2 y_0$ 的值代入式(2.10.4)，得到迭代公式

$$t^{(k)} = 0.5443313 - 0.04943527t^{(k-1)}(t^{(k-1)} - 1) \quad (k = 1,2,\cdots),$$

由此公式算出

$$t^{(1)} = 0.5565926, \quad t^{(2)} = 0.5565318, \quad t^{(3)} = 0.5565321.$$

从 $t^{(3)}$ 得到

$$\begin{aligned} \text{sinarc}0.9800000 \approx x^{(3)} &= x_0 + t^{(3)}h \\ &= 1.76 + 0.5565321 \times 0.02 = 1.771131. \end{aligned}$$

这个值准确到第六位小数.

2.10.3 构造反函数的插值多项式

设函数 f 在点 $x_0, x_1, \cdots, x_n$ 上的值为 $y_0, y_1, \cdots, y_n$. 若 f 的反函数 f^{-1} 存在，则 f^{-1} 在点 $y_0, y_1, \cdots, y_n$ 上的值为 $x_0, x_1, \cdots, x_n$，这样，反函数 f^{-1} 以 $y_0, y_1, \cdots, y_n$ 为节点的插值多项式为

$$\begin{aligned} p(y) = & f^{-1}(y_0) + f^{-1}[y_0, y_1](y - y_0) + \\ & f^{-1}[y_0, y_1, y_2](y - y_0)(y - y_1) + \cdots + \\ & f^{-1}[y_0, y_1, \cdots, y_n](y - y_0)(y - y_1)\cdots(y - y_{n-1}). \end{aligned} \tag{2.10.5}$$

由于这是利用 f 的离散数据来建立 f^{-1} 的插值公式，$y_0, y_1, \cdots, y_n$ 一般不会是等距分布，因此上面写出的是均差形式的公式.

对 $y_0, y_1, \cdots, y_n$ 确定的区间内的任何值 c，可由式(2.10.5)算出 $f^{-1}(c)$ 的近似值，或者算出方程 $f(x)=c$ 的近似解.

例 2.10.2 设 $f(x)=x^3-3x^2-x+9$，已知 f 的下列函数值：

x	-1.3	-1.4	-1.5	-1.6	-1.7
$f(x)$	3.033	1.776	0.375	-1.176	-2.883

求方程 $f(x)=0$ 在区间$[-1.7,-1.3]$上的根的近似值.

解 f 的反函数 f^{-1} 的均差如下：

y_i	$f^{-1}(y_i)$	$f[x_0,x_i]$	$f[x_0,x_1,x_i]$	$f[x_0,x_1,x_2,x]_i$	$f[x_0,\cdots,x_4]$
3.033	−1.3				
1.776	−1.4	0.0795545			
0.375	−1.5	0.0752445	0.0030763		
−1.176	−1.6	0.0712758	0.0028044	0.0001753	
−2.883	−1.7	0.0676133	0.0025630	0.0001575	0.0000104

由均差表得出 f^{-1} 的插值多项式为

$$\begin{aligned}p(y) =& -1.3+0.0795545(y-3.033)+\\&0.0030763(y-3.033)(y-1.776)+\\&0.0001753(y-3.033)(y-1.776)(y-0.375)+\\&0.0000104(y-3.033)(y-1.776)(y-0.375)\times\\&(y+1.176)\end{aligned}$$

经计算可得方程 $f(x)=0$ 在$[-1.7,-1.3]$上的根 x 的近似值

$$x=f^{-1}(0)\approx p(0)=-1.525097.$$

根 x 的实际七位准确值是-1.525102.

2.11 有理函数插值

2.11.1 有理插值的存在惟一性

假设给定 $m+n+1$ 个互异的点 $x_0,x_1,\cdots,x_{m+n}$和相应的函数值

$$f(x_0),f(x_1),\cdots,f(x_{m+n}),$$

构造一个有理分式函数

$$\begin{aligned}R_{mn}(x)&=\frac{N_m(x)}{D_n(x)}\\&=\frac{a_mx^m+a_{m-1}x^{m-1}+\cdots+a_1x+a_0}{b_nx^n+b_{n-1}x^{n-1}+\cdots+b_1x+b_0},\end{aligned}\qquad(2.11.1)$$

使之满足插值条件

$$R_{mn}(x_j) = f(x_j) \quad (j = 0,1,\cdots,m+n). \quad (2.11.2)$$

这种问题就是所谓**有理函数插值**问题. 当 $n=0$ 时, R_{mn} 是一个 m 次多项式, 插值问题(2.11.2)的解存在惟一. 但当 $n>0$ 时, R_{mn} 是一个真正的有理分式函数, 插值问题(2.11.2)的解不是存在惟一的. 例如, 假定 $m=0, f(x_j)=0, f(x_k)\neq 0$ 这种特例, 此时

$$R_{0n}(x) = \frac{a_0}{b_n x^n + \cdots + b_1 x + b_0},$$

由 $R_{0n}(x_j)=0$ 可推知 $a_0=0$, 但 $a_0=0$ 时 $R_{0n}(x_k)\neq 0$ 显然不成立, 故这个插值问题无解. 这就说明满足插值条件(2.11.2)的有理分式函数的解是否存在是有条件的. 但可以证明**插值问题(2.11.2)的解如果存在则必惟一**. 这里惟一的概念是指两个有理分式

$$R_1(x) = \frac{P_1(x)}{Q_1(x)}, \quad R_2(x) = \frac{P_2(x)}{Q_2(x)}$$

满足条件

$$P_1(x)Q_2(x) \equiv P_2(x)Q_1(x).$$

此时, 称 $R_1(\cdot)$ 与 $R_2(\cdot)$ **等价**, 并用记号 $R_1(\cdot)\sim R_2(\cdot)$ 表示. 如果存在常数 $\alpha\neq 0$, 使

$$P_2(x) = \alpha P_1(x), \quad Q_2(x) = \alpha Q_1(x),$$

则称 $R_1(\cdot)$ 与 $R_2(\cdot)$ **恒等**, 记作 $R_1(\cdot)\equiv R_2(\cdot)$, 两个有理分式 $R_1(\cdot)$ 与 $R_2(\cdot)$ 等价, 必须而且只需 $R_1(\cdot)$ 和 $R_2(\cdot)$ 的最简分式 $\bar{R}_1(\cdot)$ 与 $\bar{R}_2(\cdot)$ 恒等. 因此, 在使用时只要两个有理分式等价, 则认为它们是同一有理分式而不加区别. 有理函数插值的惟一性即建立在这种意义上.

因此, 对有理函数插值来说, 关键问题是存在性和具体解法. 当有理分式(2.11.1)满足插值条件(2.11.2)时, 只要分母 $D_n(x_j)\neq 0$ $(j=0,1,\cdots,m+n)$, 就有

$$N_m(x_j) - f(x_j)D_n(x_j) = 0 \quad (j = 0,1,\cdots,m+n), \quad (2.11.3)$$

它是关于系数 $a_m,\cdots,a_0,b_n,\cdots,b_0$ 的线性方程组. 这里未知数约去一个常数后实质上只有 $m+n+1$ 个,与方程个数相同,方程(2.11.3)比非线性方程(2.11.2)容易求解,但它们是否等价是有条件的.

定理 2.11.1 若线性方程组(2.11.3)有非平凡解,为使满足插值条件(2.11.2)的最简有理分式 $R_{mn}(x)=p_m(x)/q_n(x)$ 存在,必须且只需(2.11.3)的任一非平凡解 $N_m^*(x)$ 或 $D_n^*(x)$ 在约去一切公共因子后所得的互质多项式 $A(x)$, $B(x)$ 仍然是(2.11.3)的解,即

$$A(x_j)-f(x_j)B(x_j)=0 \quad (j=0,1,\cdots,m+n).$$

这个定理给出了方程(2.11.2)与(2.11.3)等价的充分必要条件,但是所给条件不便于检验. 定理 2.11.2 给出的是一个便于应用的条件.

定理 2.11.2 设 (x_j,y_j) 中各 $x_j\,(j=0,1,\cdots,m+n)$ 是互异的, $y_j=f(x_j)\,(j=0,1,\cdots,m+n)$. 为使满足插值条件(2.11.2)的最简有理分式

$$R_{mn}(x)=\frac{N_m(x)}{D_n(x)}$$

存在,必须且只需下述各矩阵

$$\boldsymbol{A}_j=\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^{m-1} & y_0 & x_0y_0 & \cdots & x_0^{n-1}y_0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{j-1} & x_{j-1}^2 & \cdots & x_{j-1}^{m-1} & y_{j-1} & x_{j-1}y_{j-1} & \cdots & x_{j-1}^{n-1}y_{j-1} \\ 1 & x_{j+1} & x_{j+1}^2 & \cdots & x_{j+1}^{m-1} & y_{j+1} & x_{j+1}y_{j+1} & \cdots & x_{j+1}^{n-1}y_{j+1} \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{m+n} & x_{m+n}^2 & \cdots & x_{m+n}^{m-1} & y_{m+n} & x_{m+n}y_{m+n} & \cdots & x_{m+n}^{n-1}y_{m+n} \end{pmatrix},$$

$$j=0,1,\cdots,m+n, \tag{2.11.4}$$

都是非奇异的.

例 2.11.3 给定$(x_0,y_0)=(0,1)$,$(x_1,y_1)=(1,0)$和$(x_2,y_2)=(2,0)$,用形如

$$R_{11}(x)=\frac{a_0+a_1x}{b_0+b_1x}$$

的有理函数对上述三个型值点插值.由于

$$\mathbf{A}_0=\begin{bmatrix}1 & 0\\ 1 & 0\end{bmatrix}$$

是一个奇异阵,由定理 2.11.2 知上述插值问题的解不存在.

2.11.2 蒂埃勒倒差商算法

蒂埃勒(Thiele)根据多项式插值的牛顿差商插值公式(见 2.4.2 节)的思想,为解决有理函数插值问题(2.11.2)设计了一种连分式插值的蒂埃勒方法,其表达式为

$$R(x)=a_0+\cfrac{x-x_0}{a_1+\cfrac{x-x_1}{a_2+\ddots\,a_{m+n-1}+\cfrac{x-x_{m+n-1}}{a_{m+n}}}}.\tag{2.11.5}$$

上式可简写为

$$R(x)=a_0+\frac{x-x_0}{a_1}+\frac{x-x_1}{a_2}+\cdots+\frac{x-x_{m+n-1}}{a_{m+n}}.$$

按插值条件(2.11.2),此时表示为

$$R(x_i)=f(x_i)\quad(i=0,1,\cdots,m+n).\tag{2.11.6}$$

由此条件可求出式(2.11.5)中的系数$a_k(k=0,1,\cdots,m+n)$的表达式,类似差商(即均差)定义可给出倒差商定义.

定义 2.11.4 给定点集$\{x_i,i=0,1,\cdots\}$,如果函数序列满足关系:

$$
\begin{cases} v_0(x) = f(x), \\ v_k(x) = \dfrac{x - x_{k-1}}{v_{k-1}(x) - v_{k-1}(x_{k-1})} \quad (k = 1,2,\cdots), \end{cases} \tag{2.11.7}
$$

称 $v_k(x)$ 为函数 $f(x)$ 在点集 $\{x_i, i=0,1,\cdots\}$ 上的 k 阶**倒差商**(inverse divided difference).

当 $k=1$ 时,$v_1(x)=\dfrac{x-x_0}{v_0(x)-v_0(x_0)}=\dfrac{x-x_0}{f(x)-f(x_0)}$,

当 $k=2$ 时,$v_2(x)=\dfrac{x-x_1}{v_1(x)-v_1(x_1)}$,

$\vdots$

式(2.11.7)可改写为

$$
v_0(x) = v_0(x_0) + \frac{x - x_0}{v_1(x)},
$$

$$
v_k(x) = v_k(x_k) + \frac{x - x_k}{v_{k+1}(x)} \quad (k = 1,2,\cdots). \tag{2.11.8}
$$

利用公式(2.11.8)可将 $f(x)$ 展开为连分式

$$
\begin{aligned}
f(x) = v_0(x) &= v_0(x_0) + \cfrac{x - x_0}{v_1(x_1) + \cfrac{x - x_1}{v_2(x)}} \\
&= v_0(x_0) + \frac{x - x_0}{v_1(x_1)} + \frac{x - x_1}{v_2(x_2)} + \cdots + \\
&\quad \frac{x - x_{m+n-1}}{v_{m+n-1}(x_{m+n})} + \frac{x - x_{m+n}}{v_{m+n}(x)}.
\end{aligned}
$$

上式右端略去最后一项 $\dfrac{x-x_{m+n}}{v_{m+n}(x)}$,并记

$$
R(x) = v_0(x_0) + \frac{x - x_0}{v_1(x_1)} + \frac{x - x_1}{v_2(x_2)} + \cdots + \frac{x - x_{m+n-1}}{v_{m+n}(x_{m+n})}, \tag{2.11.9}
$$

这与式(2.11.5)的连分式形式相同,假设对 $x_0, x_1, \cdots, x_{m+n}$ 倒差商 $v_k(x_k)$,$k=0,1,\cdots,m+n$ 有定义,由式(2.11.9),

当 $x=x_0$ 时，$R(x_0)=v_0(x_0)=f(x_0)$，

当 $x=x_1$ 时，$R(x_1)=v_0(x_0)+\dfrac{x_1-x_0}{v_1(x_1)}=v_0(x_1)=f(x_1)$，

当 $x=x_2$ 时，$R(x_2)=v_0(x_0)+\cfrac{x_2-x_0}{v_1(x_1)+\cfrac{x_2-x_1}{v_2(x_2)}}$

$$=v_0(x_0)+\frac{x_2-x_0}{v_1(x_2)}=v_0(x_2)=f(x_2).$$

一般情形可用归纳法证明 $R(x_i)=f(x_i)$ 对 $i=0,1,\cdots,m+n$ 成立. 故式(2.11.9)得到的 $R(x)$ 即为满足插值条件(2.11.6)的有理函数插值. 求 $R(x)$ 时只需计算倒差商 $v_k(x_k)$，通常可构造倒差商表(见表 2.7).

表 2.7

x_i	$f(x_i)=v_0(x_i)$	$v_1(x_i)$	$v_2(x_i)$	$v_3(x_i)$	$\cdots$	$v_{m+n}(x_i)$
x_0	$f_0=\underline{v_0(x_0)}$					
x_1	$f_1=v_0(x_1)$	$\underline{v_1(x_1)}$				
x_2	$f_2=v_0(x_2)$	$v_1(x_2)$	$\underline{v_2(x_2)}$			
x_3	$f_3=v_0(x_3)$	$v_1(x_3)$	$v_2(x_3)$	$\underline{v_3(x_3)}$		
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	
x_{m+n}	$f_{m+n}=v_0(x_{m+n})$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\underline{v_{m+n}(x_{m+n})}$

表中的倒差商可由以下公式计算：

$$v_k(x_i)=\frac{x_i-x_{k-1}}{v_{k-1}(x_i)-v_{k-1}(x_{k-1})}$$

$$(k=1,2,\cdots;\quad i=k,k+1,\cdots,m+n).$$

例 2.11.5 给出函数表

x_i	0	1	2	3	4
f_i	1	1/2	1/5	1/10	1/17

求有理插值 $R(x)$.

解 先构造倒差商表

x_i	$f_i=v_0(x_i)$	$v_1(x_i)$	$v_2(x_i)$	$v_3(x_i)$	$v_4(x_i)$
0	$\underline{1}$				
1	1/2	$\underline{-2}$			
2	1/5	−2/5	$\underline{-2}$		
3	1/10	−10/3	−3/2	$\underline{2}$	
4	1/17	−17/4	−4/3	3	$\underline{1}$

表中对角元素的数值即为公式(2.11.9)用到的倒差商，由式(2.11.9)可得有理插值函数为

$$R(x)=1+\frac{x-0}{-2}\,\underset{+}{}\frac{x-1}{-2}\,\underset{+}{}\frac{x-2}{2}\,\underset{+}{}\frac{x-3}{1}$$

$$=1+\cfrac{x}{-2+\cfrac{x-1}{-2+\cfrac{x-2}{x-1}}}$$

$$=1+\cfrac{x}{-2+\cfrac{(x-1)^2}{-x}}$$

$$=\frac{1}{1+x^2}.$$

3 函数逼近与曲线拟合

3.1 函数空间的范数与最佳逼近问题

3.1.1 函数逼近与函数空间的范数

在区间$[a,b]$上给定一个连续函数$f(\cdot)$,用简单函数$p(\cdot)$去近似$f(\cdot)$,就是函数逼近要研究的问题.通常$p(\cdot)$可取为多项式、有理分式或三角多项式,但由于计算机上只能做四则运算,所以,如果要利用逼近函数计算$f(x)$的值,则取多项式与有理分式逼近,其实际意义更大.度量逼近误差的标准可以各不相同,但比较重要的只有两种,其对应的函数逼近称为一致逼近与平方逼近.

在区间$[a,b]$上的所有连续函数集合记为$C[a,b]$称为连续函数空间,它也是一个线性空间,按线性空间范数定义,可给出下面定义.

定义 3.1.1 设$f\in C[a,b]$,若存在惟一实数$\|\cdot\|$,满足条件:

(1) $\|f\|\geqslant 0$,当且仅当$f=0$时$\|f\|=0$;

(2) $\|\alpha f\|=|\alpha|\,\|f\|$,其中$\alpha\in\mathbb{R}$;

(3) $\|f+g\|\leqslant\|f\|+\|g\|,\quad\forall f,g\in C[a,b]$.

则称$\|\cdot\|$为连续函数空间$C[a,b]$的**范数**(norm).

常用的范数有以下三种:

$\|f\|_\infty=\max\limits_{a\leqslant x\leqslant b}|f(x)|$,称为$\infty$-范数或最大范数.

$\|f\|_1=\int_a^b|f(x)|\,\mathrm{d}x$,称为1-范数.

$\| f \|_2 = \left(\int_a^b f^2(x)\mathrm{d}x\right)^{1/2}$，称为 2- 范数.

容易验证上述三种常用范数均满足范数定义中的三个条件.

3.1.2 最佳逼近问题

考虑连续函数空间 $C[a,b]$上的元素 $f(x)$，即 $f\in C[a,b]$，若 $C[a,b]$的子集 Φ_n 表示为

$$\Phi_n = \mathrm{span}\{\varphi_0, \varphi_1, \cdots, \varphi_n\}, \tag{3.1.1}$$

其中 $\varphi_i \in C[a,b](i=0,1,\cdots,n)$，且元素 $\varphi_0, \varphi_1, \cdots, \varphi_n$ 线性无关，即找不到不全为零的常数 $a_i(i=0,1,\cdots,n)$，使得

$$a_0\varphi_0(x) + a_1\varphi_1(x) + \cdots + a_n\varphi_n(x) = 0, \quad \forall x \in \mathbb{R}$$

成立. 由式(3.1.1)知，对 $\forall \varphi \in \Phi_n$ 可表示为

$$\varphi(x) = a_0\varphi_0(x) + a_1\varphi_1(x) + \cdots + a_n\varphi_n(x). \tag{3.1.2}$$

今后称 $\varphi(x)$为**广义多项式**. 若取

$$\varphi_0(x) = 1, \quad \varphi_1(x) = x, \quad \cdots, \quad \varphi_n(x) = x^n,$$

显然 $\varphi_i \in C[a,b](i=0,1,\cdots,n)$且$\{\varphi_i = x^i\}_0^n$. 线性无关，此时子集记为 $H_n = \mathrm{span}\{1,x,\cdots,x^n\}$表示次数小于等于 n 的多项式

$$p(x) = a_0 + a_1x + \cdots + a_nx^n \tag{3.1.3}$$

所构成的集合，它是 $C[a,b]$的一个子空间.

定义 3.1.2 设 $f\in C[a,b]$为给定函数，$\Phi_n \subset C[a,b]$，称

$$\Delta(f,\Phi_n) = \inf_{\varphi\in\Phi_n} \| f - \varphi \| \tag{3.1.4}$$

为子集 Φ_n 对函数 f 的最佳逼近. 而使式(3.1.4)成立的元素 $\varphi^* \in \Phi_n$，即

$$\Delta(f,\varphi^*) = \| f - \varphi^* \| = \Delta(f,\Phi_n) = \inf_{\varphi\in\Phi_n} \| f - \varphi \|,$$

称 $\varphi^*(x)$为 $f(x)$的**最佳逼近广义多项式**.

当范数 $\|\cdot\|$ 取为 $\|\cdot\|_\infty$ 时，称为**最佳一致逼近**，当 $\|\cdot\|$ 取为 $\|\cdot\|_2$ 时，称为**最佳平方逼近**.

3.2 最佳一致逼近

3.2.1 连续函数的一致逼近

考虑在区间$[a,b]$上用多项式 $p_n \in H_n$ 一致逼近连续函数 $f \in C[a,b]$，如果成立

$$\lim_{n\to\infty} \| f - p_n \|_\infty = \lim_{n\to\infty} \max_{a\leqslant x\leqslant b} | f(x) - p_n(x) | = 0,$$

则称 $p_n(x)$在$[a,b]$上一致收敛于 $f(x)$. p_n 称为**一致逼近多项式.**

关于空间 $C[a,b]$ 中的一致逼近问题，维尔斯特拉斯(Weierstrass)在 1885 年首先证明了下述著名定理.

定理 3.2.1 设 $f \in C[a,b]$，对任意给定的 $\varepsilon>0$，总存在一个代数多项式 $p_n(x)$，使得

$$\| f - p_n \|_\infty = \max_{a\leqslant x\leqslant b} | f(x) - p_n(x) | < \varepsilon.$$

定理 3.2.1 说明对任意一个 $C[a,b]$上的连续函数都可找到一串代数多项式去逼近，使其达到任何事先指定的精度. 这个定理有很多不同证法，但公认的最完美的证法是构造伯恩斯坦(Бернштейи)多项式

$$B_n(f,x) = \sum_{k=0}^{n} \binom{n}{k} f\left(\frac{k}{n}\right) x^k (1-x)^{n-k} \quad (0 \leqslant x \leqslant 1), \tag{3.2.1}$$

其中$\binom{n}{k} = \dfrac{n(n-1)\cdots(n-k+1)}{k!}$，取$[a,b]=[0,1]$，$p_n(x) = B_n(f,x)$，即可证明$\lim\limits_{n\to\infty} p_n(x) = f(x)$对 $\forall x \in [0,1]$一致成立. 这就表明任何连续函数总存在一致逼近多项式，但用 $f(x)$的伯恩斯坦多项式逼近 $f(x)$，收敛很慢. 如果 $f(x)$在$[a,b]$上的 $n+1$ 阶导数连续，取 $x_0 \in (a,b)$，则由泰勒展开可知

$$f(x)=f(x_0)+f'(x_0)(x-x_0)+\cdots+$$
$$\frac{1}{n!}f^{(n)}(x_0)(x-x_0)^n+R_{n+1}(x).$$

记它的前 n 项部分和为 $p_n(x)$，则有

$$p_n(x)=f(x_0)+f'(x_0)(x-x_0)+\cdots+$$
$$\frac{1}{n!}f^{(n)}(x_0)(x-x_0)^n, \tag{3.2.2}$$

$$R_{n+1}(x)=\frac{1}{(n+1)!}f^{(n+1)}(\xi)(x-x_0)^{n+1}$$
$$(\xi\text{在 }x\text{ 与 }x_0\text{ 之间}). \tag{3.2.3}$$

由于对 $\forall x\in[a,b]$，有

$$\lim_{n\to\infty}|R_{n+1}(x)|=0,$$

即 $\lim\limits_{n\to\infty}\|f-p_n\|_\infty=0$，故 $p_n(x)$ 是 $f(x)$ 的一致逼近多项式.

例 3.2.2 对 $f(x)=e^x$，用泰勒展开求$[-1,1]$上的四次一致逼近多项式.

解 取 $x_0=0$，得

$$p_4(x)=1+x+\frac{1}{2}x^2+\frac{1}{6}x^3+\frac{1}{24}x^4,$$

$$R_5(x)=e^x-p_4(x)=\frac{e^\xi}{120}x^5\quad(\xi\text{在 }x\text{ 与 }0\text{ 之间}),$$

$$\|e^x-p_4(x)\|_\infty\leqslant\frac{e}{120}\approx 0.0226.$$

在区间$[-1,1]$上用 $p_4(x)$ 逼近 e^x 的误差分布见图 3.1，它在 $x_0=0$ 附近误差较小，但在区间两端误差很大，表明误差分布极不均匀.

一致逼近多项式并非惟一的，而且，当精度要求较高时，多项式次数 n 可能很大，因此为求得使误差 $\|f-p_n\|_\infty$ 取得最小的逼近多项式，就要研究最佳一致逼近问题.

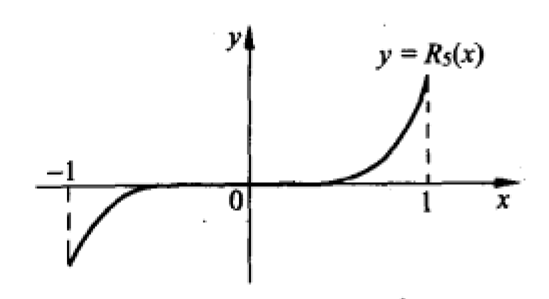


图 3.1　$p_4(x)\approx e^x$ 的误差曲线

3.2.2　最佳一致逼近多项式

在次数不超过 n 的多项式集合 H_n 中求 $p_n(\cdot)$，使它与 $f\in C[a,b]$的误差

$$\|f-p_n\|_\infty=\max_{a\leqslant x\leqslant b}|f(x)-p_n(x)|$$
$$=\inf_{p_n\in H_n}\max_{a\leqslant x\leqslant b}|f(x)-p_n(x)|$$

这就是最佳一致逼近问题.

定义 3.2.3　给定 $f\in C[a,b]$，若存在 $p_n^*\in H_n$，使

$$\Delta(f,p_n^*)=\|f-p_n^*\|_\infty$$
$$=\inf_{p_n\in H_n}\|f-p_n\|_\infty=E_n \qquad (3.2.4)$$

则称 p_n^* 是 f 在 $[a,b]$ 上的**最佳一致逼近**(minimax approximation)**多项式**，$\Delta(f,p_n^*)$称为**最佳偏差**(minimax error). 它等于最小偏差值 E_n.

理论上已证明，对任何 $f\in C[a,b]$，都存在惟一的 $p_n^*\in H_n$，使式(3.2.4)成立. 实际上在集合 H_n 中每一元素 $p_n\in H_n$ 都对应一个偏差 $\Delta(f,p_n)$，由于 $\Delta(f,p_n)=\|f-p_n\|_\infty\geqslant 0$，故集合 $\{\Delta(f,p_n)\}$有下界，从而有下确界 $E_n=\inf\limits_{p_n\in H_n}\Delta(f,p_n)$. 如果存在 $p_n^*\in H_n$ 使 $\|f-p_n^*\|_\infty=E_n$，p_n^* 就是所要求的最佳一致逼近多项式. 切比雪夫(Chebyshev)对最佳一致逼近多项式的特性，给出

了下面的重要定理.

定理 3.2.4 （切比雪夫） 设 $f\in C[a,b]$，$n\geqslant 0$. $p_n^*(\cdot)\in H_n$ 是 f 在 $[a,b]$ 上的最佳一致逼近多项式的充分必要条件是，$p_n^*(x)$ 在 $[a,b]$ 上至少有 $n+2$ 个点

$$a\leqslant x_0 < x_1 < \cdots < x_{n+1}\leqslant b,$$

使

$$p_n^*(x_k)-f(x_k)=(-1)^k\sigma\|p_n^*-f\|_\infty,$$
$$\sigma=\pm 1,k=0,1,\cdots,n+1. \qquad (3.2.5)$$

这个定理表明，最佳一致逼近多项式 $p_n^*(x)$ 的特性，即 $p_n^*(x)$ 逼近 $f(x)$ 的误差分布是均匀的，如图 3.2 所示.

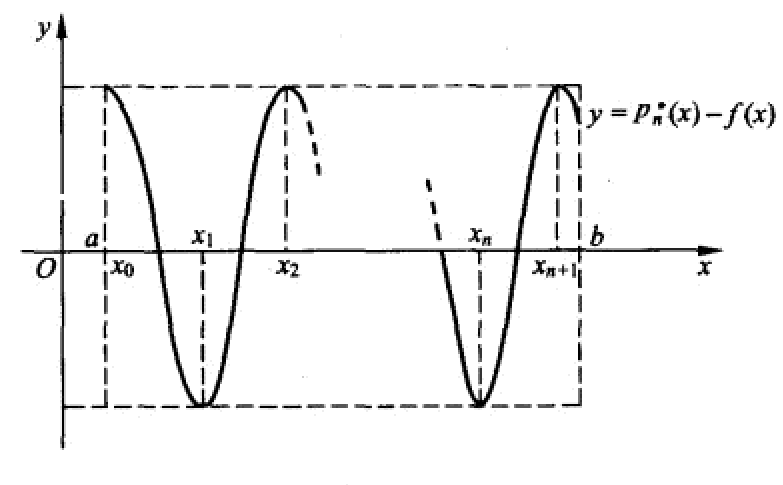


图 3.2

若在空间 $C[a,b]$ 中取子集 $\Phi_n=\text{span}\{\varphi_0,\varphi_1,\cdots,\varphi_n\}$. 由定义 3.1.2 知，若存在 $\varphi^*\in\Phi_n$ 使

$$\|f-\varphi^*\|_\infty=\inf_{\varphi\in\Phi_n}\|f-\varphi\|_\infty,$$

则称 $\varphi^*(x)$ 为 $f\in C[a,b]$ 的最佳一致逼近广义多项式，对于代数多项式集合 H_n 的元素 $p_n(x)$，在 $[a,b]$ 上最多只能有 n 个不同的零点，根据切比雪夫定理 3.2.4 知最佳一致逼近多项式 $p_n^*(x)$ 有 $n+2$ 个轮流为"+"、"−"的偏差点，对广义多项式 $\varphi\in\Phi_n$ 也要求

在$[a,b]$上至多具有 n 个不同的零点，因此要对广义多项式引进更广泛的哈尔(Haar)条件.

定义 3.2.5 函数 $\varphi_i(x)\in C[a,b](i=0,1,\cdots,n)$线性无关，若子集 $\Phi_n=\text{span}\{\varphi_0,\varphi_1,\cdots,\varphi_n\}\subset C[a,b]$中任一不恒为零的广义多项式(3.1.2)，即

$$\varphi(x)=a_0\varphi_0(x)+a_1\varphi_1(x)+\cdots+a_n\varphi_n(x)$$

在区间$[a,b]$上至多具有 n 个不同的零点，则称函数 $\varphi_i(x)(i=0,1,\cdots,n)$在$[a,b]$上满足哈尔条件，也可称子集 Φ_n 满足哈尔条件.

显然，子集 H_n 是满足哈尔条件的.

有了定义 3.2.5，就可类似定理 3.2.4 得到下面定理.

定理 3.2.6 若子集 $\Phi_n=\text{span}\{\varphi_0,\varphi_1,\cdots,\varphi_n\}$满足 Haar 条件则对任意给定的函数 $f\in C[a,b]$，使广义多项式

$$\varphi^*(x)=\sum_{i=0}^{n}a_i^*\varphi_i(x)\in\Phi_n$$

成为函数 $f(x)$在空间 $C[a,b]$上的最佳(一致)逼近广义多项式的充分必要条件是 $\varphi^*(x)$在$[a,b]$上至少存在 $n+2$ 个轮流为“+”、“-”的偏差点，即

$$\varphi^*(x_i)-f(x_i)=(-1)^i\sigma\|f-\varphi_n^*\|_\infty,$$
$$\sigma=\pm1,i=0,1,\cdots,n+1.$$

利用定理 3.2.6 还可证明当子集 $\Phi_n\subset C[a,b]$满足 Haar 条件时，则对任给的函数 $f\in[a,b]$的最佳逼近广义多项式是惟一的.

Haar 还提出了下面的最佳逼近广义多项式的惟一性定理.

定理 3.2.7 对任何函数 $f\in C[a,b]$，子集 $\Phi_n\subset C[a,b]$中存在惟一的最佳逼近广义多项式的充分必要条件是子集 $\Phi_n\subset C[a,b]$满足 Haar 条件.

3.3 最佳一致逼近多项式的数值方法

3.3.1 最佳一致线性逼近

关于求给定函数 $f(x)$ 的最佳一致逼近多项式，理论上可由定理 3.2.4 通过方程组(3.2.5)求得，但由于它的非线性性质，求解是十分困难的，通常只能对 $n=1$ 给出具体算法，对 $n\geqslant 2$ 的情形通常可用逐次逼近的列梅兹(Remes)算法. 对 $n=1$ 的算法如下：

假定 $f(\cdot)$ 在 $[a,b]$ 上二阶导数 $f''(\cdot)$ 连续，且 $f''(\cdot)$ 不变号，设 $p_1^*(x)=a_0+a_1x$，由定理 3.2.4 可知，在区间 $[a,b]$ 上至少有 3 个点 $a\leqslant x_0<x_1\leqslant x_2\leqslant b$，使

$$p_1^*(x_k)-f(x_k)=(-1)^k\sigma\|p_1^*-f\|_\infty,\quad \sigma=\pm 1, k=0,1,2. \tag{3.3.1}$$

由于 $f''(x)$ 在 $[a,b]$ 上不变号，故 $f'(x)$ 在 $[a,b]$ 单调，所以，$[p_1^*(x)-f(x)]'=a_1-f'(x)=0$ 只能有一个解，记作 x_1，即 $a_1=f'(x_1)$；另两点必在区间端点，即 $x_0=a$，$x_2=b$，于是由式(3.3.1)得

$$\begin{aligned}p_1^*(a)-f(a)&=-[p_1^*(x_1)-f(x_1)]\\&=p_1^*(b)-f(b),\end{aligned}$$

由此可得

$$\begin{cases}a_1=\dfrac{f(b)-f(a)}{b-a}=f'(x_1),\\ a_0=\dfrac{f(a)+f(x_1)}{2}-a_1\dfrac{a+x_1}{2}.\end{cases} \tag{3.3.2}$$

例 3.3.1 求 $f(x)=e^x$ 在 $[-1,1]$ 上的一次最佳一致逼近 $p_1^*(x)\in H_1$（如图 3.3 所示）.

解 令 $p_1^*(x)=a_0+a_1x$，由式(3.3.2)得

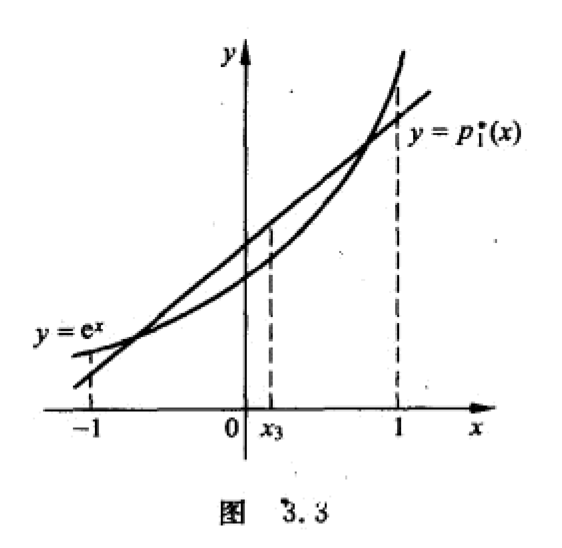


图 3.3

$$\begin{cases}a_1=\dfrac{e^1-e^{-1}}{2}\approx 1.1752,\\ f'(x_1)=e^{x_1}=a_1\approx 1.1752\rightarrow x_1=\ln a_1\approx 0.1614,\\ a_0=\dfrac{e^{-1}+a_1}{2}+\dfrac{a_1}{2}(1-x_1)\approx 1.2643.\end{cases}$$

于是得到最佳一致线性逼近

$$p_1^*(x)=1.2643+1.1752x,\quad \Delta(f,p_1^*)\approx 0.2788.$$

3.3.2 列梅兹算法

切比雪夫定理 3.2.4 从理论上给出了求最佳一致逼近多项式 $p_n^*(x)$的计算方法，但由于它的非线性性质，求解是很困难的. 列梅兹于 1957 年采用线性的逐次逼近方法，由于收敛很快，效果较好，是求解最佳一致逼近多项式的有效算法. 下面给出具体算法.

设 $f\in C[a,b]$，其最佳一致逼近多项式

$$p_n^*(x)=\sum_{k=0}^{n}a_k^*x^k$$

的 $n+1$ 个系数 $a_k^*(k=0,1,\cdots,n)$，最小偏差 E_n 和 $n+2$ 个偏差

点 $a \leqslant x_0^* < x_1^* < \cdots < x_{n+1}^* \leqslant b$，一共为 $2n+4$ 个未知数. 由式(3.2.5)得

$$\begin{cases} \sum_{j=0}^{n} a_j^* x_k^j - f(x_k^*) = (-1)^k \sigma E_n, \\ \qquad \sigma = \pm 1; k = 0,1,\cdots,n+1, \\ (x_0^* - a)(x_{n+1}^* - b)[p_n^{*\prime}(x_k^*) - f'(x_k^*)] = 0, \\ \qquad k = 0,1,\cdots,n+1. \end{cases} \tag{3.3.3}$$

这是一个 $2n+4$ 个未知数的非线性方程组，一般情况是很难求解的，为了求 $p_n^*(x)$，可由方程组(3.3.3)出发，利用切比雪夫多项式 $\mathrm{Tn}(x)$(见 3.4.3 节)，先求出近似程度好的偏差点，再用逐次逼近方法求出 $p_n^*(x)$，这就是列梅兹**算法**(algorithms of Remes). 为方便起见，仍假定 $f \in C[-1,1]$，算法分三步进行.

1. 给出 $n+2$ 个初始偏差点

$$-1 \leqslant x_0 < x_1 < \cdots < x_{n+1} \leqslant 1.$$

通常，可用 $\mathrm{T}_{n+1}(x)$ 的偏差点 $x_k = \cos\left(\frac{k\pi}{n+1}\right)(k=0,1,\cdots,n+1)$ 解 $n+2$ 个未知数 $a_0,\cdots,a_n$ 及 E 的线性方程组

$$a_0 + a_1 x_k + \cdots + a_n x_k^n - f(x_k) = (-1)^k E,$$
$$k = 0,1,\cdots,n+1,$$

从而得到逼近多项式 $p_n(x) = \sum_{k=0}^{n} a_k x^k$ 及 E.

2. 求 $n+2$ 个新的偏差点

$$-1 \leqslant z_0 < z_1 < \cdots < z_{n+1} \leqslant 1,$$

要求 $p_n(z_k) - f(z_k)$ 正负交错，且

$$p'_n(z_k) - f'(z_k) = 0 \quad (k = 1,\cdots,n),$$

也可包括 z_0 及 z_{n+1}. 如上式只有 n 个点成立，则可取 $z_0 = -1$，$z_{n+1} = 1$. 在某些点 z_k 上，满足

$$\| f - p_n \|_\infty = | f(z_k) - p_n(z_k) |.$$

3. 根据定理(3.2.4)和偏差点$\{z_k\}$的性质，有

$$\begin{aligned} m &= \min_k |f(z_k) - p_n(z_k)| \\ &\leqslant \| f - p_n \|_\infty \\ &\leqslant M = \max_k |f(z_k) - p_n(z_k)|. \end{aligned}$$

如M/m充分靠近1，则$p_n(x)$为所求，记作$p_n^*(x)$. 例如，当$M/m \leqslant 1.05$时，则得$p_n^*(x)$；否则，用点$\{z_k\}$代替$\{x_k\}$转回第1步继续迭代.

根据列梅兹算法，可求得$f(x)=e^x$在$[-1,1]$上的最佳一致逼近

$$p_3^*(x) = 0.994579 + 0.995668x + 0.542973x^2 + 0.179533x^3,$$
$$\| f - p_n^* \|_\infty = 0.00553. \tag{3.3.4}$$

$p_n^*(x)$逼近$f(x)$的误差分布见图3.4.

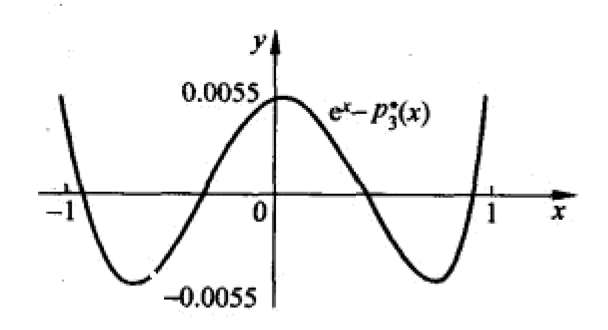


图 3.4　e^x的三次最佳逼近误差

3.4　正交多项式

3.4.1　内积与正交多项式

在讨论平方逼近之前需要先引入以下的概念.

定义 3.4.1　非负函数$\rho(x)$在区间$[a,b]$(有限或无限)上满足下列条件，则称$\rho(x)$为$[a,b]$上的**权函数**(weight function).

(1) $\int_a^b x^n \rho(x) \mathrm{d}x$ 对 $n = 0,1,\cdots$ 存在且为有限值；

(2) 对非负连续函数 f，若$\int_a^b f(x)\rho(x)\mathrm{d}x = 0$，则在$[a,b]$上 $f(x) \equiv 0$.

定义 3.4.2 给定 $f,g \in C[a,b]$，ρ 是$[a,b]$上的权函数，称

$$(f,g) = \int_a^b \rho(x) f(x) g(x) \mathrm{d}x$$

为函数 f 与 g 在$[a,b]$上关于权 $\rho(x)$ 的**内积**(inner product).

内积有下列简单性质：

(1) $(f,g)=(g,f)$；

(2) $(af,g)=a(f,g), a\in\mathbb{R}$；

(3) $(f_1+f_2,g)=(f_1,g)+(f_2,g)$；

(4) 当 $f\not\equiv 0$ 时，$(f,f)>0$.

利用内积可定义 f 的欧几里得(Euclid)**范数**(norm)

$$\| f \|_2 = \sqrt{(f,f)}.$$

它满足范数的三条基本性质：

(1) 当 $f\not\equiv 0$ 时，$\| f \|_2>0$，当且只当 $f\equiv 0$ 时 $\| f \|_2=0$；

(2) $\| af \|_2=|a| \, \| f \|_2, a\in\mathbb{R}$；

(3) $\| f+g \|_2 \leqslant \| f \|_2+\| g \|_2$(三角不等式).

此外，还有下列结论：

(1) $|(f,g)| \leqslant \| f \|_2 \| g \|_2$，称为柯西-施瓦茨(Cauchy-Schwartz)不等式；

(2) $\| f+g \|_2^2+\| f-g \|_2^2=2\| f \|_2^2+2\| g \|_2^2$，此即平行四边形定律.

定义 3.4.3 若内积

$$(f,g) = \int_a^b \rho(x) f(x) g(x) \mathrm{d}x = 0,$$

则称 f 与 g 在区间$[a,b]$上带权 $\rho(x)$ **正交**(orthogonal)，若函数族

$\varphi_0,\varphi_1,\cdots,\varphi_n,\cdots$满足

$$(\varphi_i,\varphi_j)=\int_a^b\rho(x)\varphi_i(x)\varphi_j(x)\mathrm{d}x$$
$$=\begin{cases}0, & i\neq j,\\ A_j>0, & i=j,\end{cases}\tag{3.4.1}$$

则称$\{\varphi_k\}$是$[a,b]$上带权$\rho(x)$的**正交函数族**(orthogonal systems of functions).

例 3.4.4 三角函数族

$$1,\cos x,\sin x,\cos 2x,\sin 2x,\cdots$$

在区间$[-\pi,\pi]$上正交. 因为

$$\begin{cases}\displaystyle\int_{-\pi}^{\pi}\begin{Bmatrix}\cos kx\\ \sin kx\end{Bmatrix}\begin{Bmatrix}\cos jx\\ \sin jx\end{Bmatrix}\mathrm{d}x=0, & k\neq j\\ \displaystyle\int_{-\pi}^{\pi}\cos kx\sin kx\,\mathrm{d}x=0,\\ \displaystyle\int_{-\pi}^{\pi}\sin^2 kx\,\mathrm{d}x=\int_{-\pi}^{\pi}\cos^2 kx\,\mathrm{d}x=\pi, & k=1,2,\cdots\end{cases}$$

定义 3.4.5 若$\varphi_0,\varphi_1,\cdots,\varphi_n$在$[a,b]$上连续,如果

$$a_0\varphi_0(x)+\cdots+a_n\varphi_n(x)=0$$

当且仅当$a_0=\cdots=a_n=0$才成立,就称$\varphi_0,\varphi_1,\cdots,\varphi_n$在$[a,b]$上**线性无关**. 若函数族$\{\varphi_k,k=0,1,\cdots\}$中任何有限个$\varphi_k$线性无关,则称其为**线性无关函数族**. 例如,函数族$1,x,\cdots,x^n,\cdots$就是$[a,b]$上的线性无关函数族.

定理 3.4.6 $\varphi_0,\varphi_1,\cdots,\varphi_n$在$[a,b]$上线性无关的充分必要条件是$\det\boldsymbol{G}_n=0$,其中

$$\boldsymbol{G}_n=\boldsymbol{G}(\varphi_0,\cdots,\varphi_n)$$
$$=\begin{bmatrix}(\varphi_0,\varphi_0) & (\varphi_0,\varphi_1) & \cdots & (\varphi_0,\varphi_n)\\ \vdots & \vdots & & \vdots\\ (\varphi_n,\varphi_0) & (\varphi_n,\varphi_1) & \cdots & (\varphi_n,\varphi_n)\end{bmatrix}.\tag{3.4.2}$$

$\det\boldsymbol{G}_n$称为**格拉姆**(Gram)**行列式**. $\boldsymbol{G}_n$称为格拉姆矩阵.

3.4.2 勒让德多项式

当权函数 $\rho(x)\equiv 1$,区间为$[-1,1]$时,由$\{1,x,\cdots,x^n,\cdots\}$正交化得到的正交多项式称为**勒让德**(Legendre)**多项式**,可表示为

$$P_0(x) = 1,$$

$$P_n(x) = \frac{1}{2^n n!}\frac{d^n}{dx^n}[(x^2-1)^n] \quad (n=1,2,\cdots). \quad (3.4.5)$$

由于$(x^2-1)^n$是$2n$次多项式,求n阶导数得

$$P_n(x) = \frac{1}{2^n n!}(2n)\cdots(n+1)x^n + a_{n-1}x^{n-1} + \cdots + a_0,$$

其最高项系数$a_n=\dfrac{(2n)!}{2^n(n!)^2}\neq 0$. 最高项系数为1的勒让德多项式可表示为

$$\tilde{P}_n(x) = \frac{n!}{(2n)!}\frac{d^n}{dx^n}[(x^2-1)^n].$$

勒让德多项式有以下重要性质:

(1) 正交性

$$(P_n,P_m) = \int_{-1}^{1} P_n(x),P_m(x)dx = \begin{cases} 0, & m\neq n, \\ \dfrac{2}{2n+1}, & m=n. \end{cases} \quad (3.4.6)$$

(2) 奇偶性

$$P_n(-x) = (-1)^n P_n(x),$$

即n为奇数时$P_n(x)$为奇函数,n为偶数时$P_n(x)$为偶函数.

(3) 递推关系

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x) \quad (n\geqslant 1). \quad (3.4.7)$$

由$P_0(x)=1,P_1(x)=x$,利用式(3.4.7)就可推出

$$P_2(x) = (3x^2 - 1)/2,$$
$$P_3(x) = (5x^3 - 3x)/2,$$
$$P_4(x) = (35x^4 - 30x^2 + 3)/8,$$
$$P_5(x) = (63x^5 - 70x^3 + 15x)/8,$$
$$P_6(x) = (231x^6 - 315x^4 + 105x^2 - 5)/16,$$
$$\vdots$$

图 3.5 给出了前面 4 个勒让德多项式的图形.

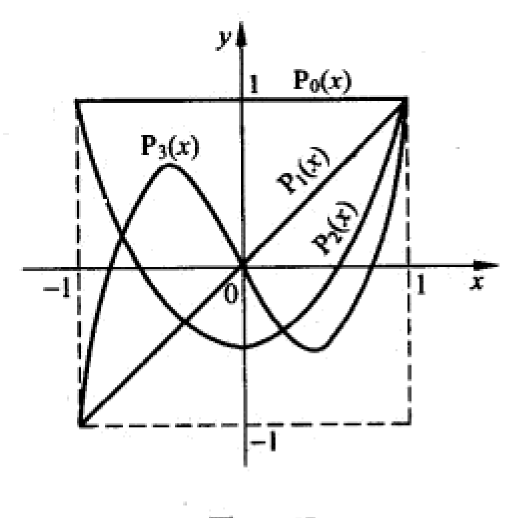


图 3.5

(4) $P_n(x)$在区间$[-1,1]$内有 n 个不同的实零点.

(5) 在所有最高项系数为 1 的 n 次多项式中,$\tilde{P}_n(x)$在区间$[-1,1]$上与零的平方误差最小.

3.4.3 切比雪夫多项式

当权函数 $\rho(x)=\dfrac{1}{\sqrt{1-x^2}}$,区间为$[-1,1]$时,得到的正交多项式就是切比雪夫多项式,它可表示为

$$T_n(x) = \cos(n\arccos x) \quad (|x| \leqslant 1). \tag{3.4.8}$$

令 $x=\cos\theta$,则 $T_n(x)=\cos n\theta$,$0\leqslant\theta\leqslant\pi$,由三角公式

切比雪夫多项式有以下重要性质.

(1) 正交性：$\{T_n(x)\}$在$[-1,1]$上带权$\rho(x)=\dfrac{1}{\sqrt{1-x^2}}$正交，即

$$\int_{-1}^{1}\frac{T_n(x)T_m(x)}{\sqrt{1-x^2}}\mathrm{d}x=\begin{cases}0, & n\neq m,\\ \dfrac{\pi}{2}, & n=m\neq 0,\\ \pi, & n=m=0.\end{cases}\quad(3.4.10)$$

(2) 极性：在所有最高项系数为 1 的多项式中，切比雪夫多项式$\widetilde{T}_n(x)=\dfrac{1}{2^{n-1}}T_n(x)$在区间$[-1,1]$上与零偏差最小，且其偏差为$\dfrac{1}{2^{n-1}}$，即

$$\begin{aligned}\frac{1}{2^{n-1}}&=\max_{-1\leqslant x\leqslant 1}|\widetilde{T}_n(x)|\\ &\leqslant\max_{-1\leqslant x\leqslant 1}|x^n+p_{n-1}(x)|\quad(p_{n-1}\in H_{n-1}).\end{aligned}$$

(3) 奇偶性：当n为奇数时$T_n(x)$为奇函数，当n为偶数时$T_n(x)$为偶函数，从而有

$$T_n(-x)=(-1)^nT_n(x).$$

(4) $T_n(x)$在区间$[-1,1]$上有n个实零点

$$x_k=\cos\frac{2k-1}{2n}\pi\quad(k=1,2,\cdots,n),$$

有$n+1$个轮流为"+"、"−"的偏差点

$$y_k=\cos\frac{k\pi}{n}\quad(k=0,1,\cdots,n).$$

(5) x^n可用$T_0,T_1,\cdots,T_n$的线性组合表示如下：

$$\begin{aligned}1&=T_0;\\ x&=T_1;\\ x^2&=\frac{1}{2}(T_0+T_2);\end{aligned}$$

$$T_{n\pm1}(x) = \cos(n\pm1)\theta = \cos(n\theta)\cos\theta \mp \sin(n\theta)\sin\theta$$

可得递推关系：

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \quad (n \geqslant 1). \qquad (3.4.9)$$

由式(3.4.8)可直接得

$$T_0(x) \equiv 1, T_1(x) = x.$$

由式(3.4.9)可推出

$$T_2(x) = 2x^2 - 1,$$
$$T_3(x) = 4x^3 - 3x,$$
$$T_4(x) = 8x^4 - 8x^2 + 1,$$
$$T_5(x) = 16x^5 - 20x^3 + 5x,$$
$$T_6(x) = 32x^6 - 48x^4 + 18x^2 - 1,$$
$$T_7(x) = 64x^7 - 112x^5 + 56x^3 - 7x,$$
$$T_8(x) = 128x^8 - 256x^6 + 160x^4 - 32x^2 + 1,$$
$$\vdots$$

由式(3.4.8)可推得 $T_n(x)$的最高项系数是 2^{n-1}. $T_n(x)$前 4 个的图形见图 3.6.

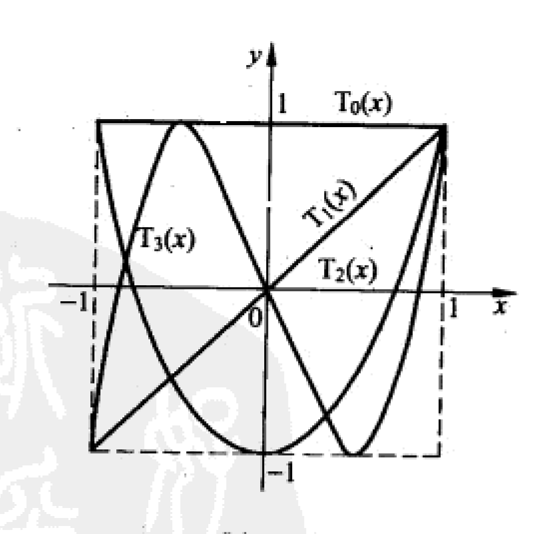


图 3.6

$$x^3=\frac{1}{4}(3T_1+T_3);$$

$$x^4=\frac{1}{8}(3T_0+4T_2+T_4);$$

$$x^5=\frac{1}{16}(10T_1+5T_3+T_5);$$

$$x^6=\frac{1}{32}(10T_0+15T_2+6T_4+T_6);$$

$$x^7=\frac{1}{64}(35T_1+21T_3+7T_5+T_7);$$

$$x^8=\frac{1}{128}(35T_0+56T_2+28T_4+8T_6+T_8).$$

3.4.4 其他常用的正交多项式

除上面两类重要的正交多项式外，常用的正交多项式还有以下三种.

1. 第二类切比雪夫多项式

在区间$[-1,1]$上取权函数 $\rho(x)=\sqrt{1-x^2}$ 得到的正交多项式就是第二类切比雪夫多项式，其表达式为

$$U_n(x)=\frac{\sin[(n+1)\arccos x]}{\sqrt{1-x^2}},\quad (|x|\leqslant 1). \tag{3.4.11}$$

令 $x=\cos\theta$ 可得

$$\int_{-1}^{1}U_n(x)U_m(x)\sqrt{1-x^2}\,\mathrm{d}x=\begin{cases}0, & m\neq n,\\ \frac{\pi}{2}, & m=n.\end{cases} \tag{3.4.12}$$

由式(3.4.11)可得递推关系

$$\begin{aligned}&U_0(x)=1,U_1(x)=2x,\\ &U_{n+1}(x)=2xU_n(x)-U_{n-1}(x)\quad (n=1,2,\cdots).\end{aligned} \tag{3.4.13}$$

2. 拉盖尔(Laguerre)多项式

在区间$[0,\infty)$上,取权函数$\rho(x)=e^{-x}$得到的正交多项式,就是拉盖尔多项式,其表达式为

$$L_n(x)=e^x\frac{d^n}{dx^n}(x^n e^{-x}). \tag{3.4.14}$$

它满足关系

$$\int_0^\infty e^{-x}L_n(x)L_m(x)dx=\begin{cases}0, & m\neq n\\(n!)^2, & m=n.\end{cases} \tag{3.4.15}$$

其递推关系为:

$$L_0(x)=1,\quad L_1(x)=1-x,$$
$$L_{n+1}(x)=(1+2n-x)L_n(x)-n^2L_{n-1}(x)\quad(n=1,2,\cdots). \tag{3.4.16}$$

3. 埃尔米特(Hermite)多项式

在区间$(-\infty,\infty)$上,取权函数$\rho(x)=e^{-x^2}$得到的正交多项式就是埃尔米特多项式,其表达式为

$$H_n(x)=(-1)^n e^{x^2}\frac{d^n}{dx^n}(e^{-x^2}), \tag{3.4.17}$$

它满足正交性条件

$$\int_{-\infty}^{+\infty}e^{-x^2}H_m(x)H_n(x)dx=\begin{cases}0, & m\neq n,\\2^n n!\sqrt{\pi}, & m=n,\end{cases} \tag{3.4.18}$$

并有递推关系

$$H_0(x)=1,\quad H_1(x)=2x,$$
$$H_{n+1}(x)=2xH_n(x)-2nH_{n-1}(x)\quad(n=1,2,\cdots). \tag{3.4.19}$$

3.5 最佳平方逼近

因为最佳一致逼近计算困难,所以在实际计算时常使用另一种度量逼近误差的标准,记作

$$\|f-r\|_2^2=\int_a^b[f(x)-r(x)]^2dx, \tag{3.5.1}$$

这里 $f\in C[a,b]$，$r\in\Phi_n$（见式(3.1.1)），用它作度量标准的函数逼近称为**平方逼近**(squares approximation).

如果在子集 Φ_n 中存在 $r_n^*(x)\in\Phi_n\subset C[a,b]$，使

$$\|f-r_n^*\|_2^2=\int_a^b[f(x)-r_n^*(x)]^2\mathrm{d}x$$

$$=\inf_{r_n\in\Phi_n}\|f-r_n\|_2^2, \qquad (3.5.2)$$

则称 $r_n^*(x)$ 是 $f(x)$ 在 $\Phi_n\subset C[a,b]$ 上的**最佳平方逼近**(best squares approximation)**广义多项式**，当 Φ_n 取为 $H_n=\mathrm{span}\{1,x,\cdots,x^n\}$ 时，

$$r_n^*(x)=a_0^*+a_1^*x+\cdots+a_n^*x^n, \qquad (3.5.3)$$

称为 $f(x)$ 在 $[a,b]$ 上的**最佳平方逼近多项式**.

例 3.5.1 对 $f(x)=\mathrm{e}^x$，$x\in[-1,1]$，在 H_1 中求最佳平方逼近多项式.

解 令 $r_1(x)=a_0+a_1x$，要求

$$\|f-r_1\|_2^2=\int_{-1}^1(\mathrm{e}^x-a_0-a_1x)^2\mathrm{d}x\equiv F(a_0,a_1)$$

最小，也就是要求 a_0^* 及 a_1^*，使 $F(a_0^*,a_1^*)\leqslant F(a_0,a_1)$，这就是二元函数求极值问题. 由极值必要条件

$$\frac{\partial F}{\partial a_0}=0,\quad \frac{\partial F}{\partial a_1}=0,$$

可得

$$\frac{\partial F}{\partial a_0}=2\int_{-1}^1(\mathrm{e}^x-a_0-a_1x)(-1)\mathrm{d}x=0,$$

$$\frac{\partial F}{\partial a_1}=2\int_{-1}^1(\mathrm{e}^x-a_0-a_1x)(-x)\mathrm{d}x=0.$$

由此方程组可解得

$$a_0=\frac{1}{2}\int_{-1}^1\mathrm{e}^x\mathrm{d}x=\frac{1}{2}(\mathrm{e}-\mathrm{e}^{-1})\approx 1.1752=a_0^*$$

$$a_1 = \frac{3}{2}\int_{-1}^{1} x\mathrm{e}^x \mathrm{d}x = 3\mathrm{e}^{-1} \approx 1.1036 = a_1^*.$$

于是在 H_1 中的最佳平方逼近多项式为

$$r_1^*(x) = 1.1752 + 1.1036x.$$

直接计算可求得

$$\| \mathrm{e}^x - r_1^* \|_\infty \approx 0.44.$$

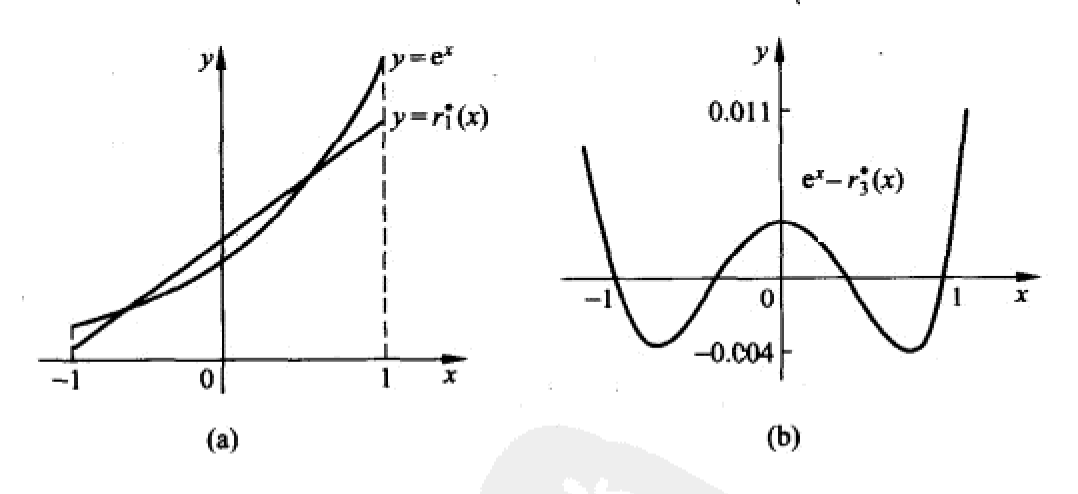
用 $r_1^*(x)$逼近 e^x 的图可见图 3.7(a).

进而可用同样方法计算 e^x 在$[-1,1]$上的三次最佳平方逼近

$$r_3^*(x) = 0.996294 + 0.997955x + 0.536722x^2 + 0.176139x^3,$$

$$\| \mathrm{e}^x - r_3^* \|_\infty \approx 0.0112. \qquad (3.5.4)$$

其误差图形可见图 3.7(b).



(a) (b)

图 3.7

下面是更一般的带权最佳平方逼近.

对给定的 $f \in C[a,b]$,$\rho(x)$为$[a,b]$上的权函数,若存在 $r_n^*(x) \in \Phi_n = \mathrm{span}\{\varphi_0, \varphi_1, \cdots, \varphi_n\}$,使

$$\left\| f - r_n^* \right\|_2^2 = \int_a^b \rho(x)[f(x) - r_n^*(x)]\mathrm{d}x$$

定义 3.4.7 设 $\varphi_n(x)$是$[a,b]$上首项系数 $a_n \neq 0$ 的 n 次多项式，$\rho(x)$为$[a,b]$上的权函数，如果多项式序列$\{\varphi_n(x)\}_{n=0}^{\infty}$满足关系式(3.4.1)，则称序列$\{\varphi_n(x)\}_{n=0}^{\infty}$在$[a,b]$上带权 $\rho(x)$正交，称$\varphi_n(x)$为$[a,b]$上带权 $\rho(x)$的 n 次正交多项式.

只要给定区间$[a,b]$及权函数 $\rho(x)$，均可由单项式序列$\{1,x,\cdots,x^n,\cdots\}$通过逐次正交化，构造出相应的正交多项式序列$\{\varphi_n(x)\}_{n=0}^{\infty}$如下：

$$\varphi_0(x)=1,\varphi_n(x)=x^n-\sum_{j=0}^{n-1}\frac{(x^n,\varphi_j)}{(\varphi_j,\varphi_j)}\varphi_j(x)\quad(n=1,2,\cdots).\tag{3.4.3}$$

由此得到的正交多项式有以下性质：

(1) $\varphi_n(x)$是最高项 x^n 的系数为 1 的 n 次多项式.

(2) 任何 n 次多项式 $p(x)\in H_n$，均可表示为 $\varphi_0(x)$，$\varphi_1(x),\cdots,\varphi_n(x)$的线性组合，即

$$p(x)=a_0\varphi_0(x)+a_1\varphi_1(x)+\cdots+a_n\varphi_n(x).$$

(3) 当 $k\neq j$ 时，$(\varphi_j,\varphi_k)=0$，且 $\varphi_k(x)$与任一次数小于 k 的多项式正交.

(4) 有递推关系

$$\varphi_{n+1}(x)=(x-\alpha_n)\varphi_n(x)-\beta_n\varphi_{n-1}(x)\quad(n=0,1,\cdots),\tag{3.4.4}$$

其中 $\varphi_0(x)=1,\varphi_{-1}(x)=0$，

$$\alpha_n=\frac{(x\varphi_n(x),\varphi_n(x))}{(\varphi_n(x),\varphi_n(x))},$$

$$\beta_n=\frac{(\varphi_n(x),\varphi_n(x))}{(\varphi_{n-1}(x),\varphi_{n-1}(x))}.$$

(5) $\varphi_n(x)(n\geqslant 1)$的 n 个零点都是在区间$[a,b]$内的单重实零点.

以下是几种常见的正交多项式.

$$= \inf_{r_n \in \Phi_n} \| f - r_n \|_2^2$$

$$= \inf_{r_n \in \Phi_n} \int_a^b \rho(x) [f(x) - r_n(x)]^2 \mathrm{d}x, \qquad (3.5.5)$$

则称 $r_n^*(x)$ 是 $f(x)$ 在 $[a,b]$ 上带权 $\rho(x)$ 的最佳平方逼近广义多项式. 当权函数 $\rho(x) \equiv 1$ 时, 就是式(3.5.2)给出的定义. 通常子集 $\Phi_n = \mathrm{span}\{\varphi_0, \varphi_1, \cdots, \varphi_n\}$ 中的坐标函数 $\varphi_k(x)(k=0,1,\cdots,n)$ 也取为 k 次多项式, 即 $\varphi_k \in H_k$, 因此 Φ_n 实质上仍为 H_n, 得到的 $r_n^*(x)$ 即为最佳平方逼近多项式. 为求出 $r_n^*(x)$ 可令 $r_n(x) = a_0\varphi_0(x) + a_1\varphi_1(x) + \cdots + a_n\varphi_n(x)$, 且 $\varphi_k \in H_k$, 记

$$F(a_0, a_1, \cdots, a_n) = \int_a^b \rho(x) \Big[f(x) - \sum_{k=0}^{n} a_k \varphi_k(x) \Big]^2 \mathrm{d}x. \qquad (3.5.6)$$

F 是 $a_0, a_1, \cdots, a_n$ 的 $n+1$ 元函数, 要求 $a_k^*(k=0,1,\cdots,n)$ 使

$$F(a_0^*, a_1^*, \cdots, a_n^*) \leqslant F(a_0, a_1, \cdots, a_n).$$

这是多元函数求最小值问题, 根据多元函数极值必要条件

$$\frac{\partial F}{\partial a_j} = 0 \quad (j = 0,1,\cdots,n)$$

即得

$$\frac{\partial F}{\partial a_j} = 2\int_a^b \rho(x) \Big[f(x) - \sum_{k=0}^{n} a_k \varphi_k(x) \Big] (-\varphi_j(x)) \mathrm{d}x$$

$$= 0 \quad (j = 0,1,\cdots,n).$$

此方程用内积记号表示为

$$\sum_{k=0}^{n} (\varphi_j, \varphi_k) a_k = (f, \varphi_j) \quad (j = 0,1,\cdots,n). \qquad (3.5.7)$$

这是关于参数 $a_0, a_1, \cdots, a_n$ 的 $n+1$ 元线性方程组, 称为**法方程**. 其系数矩阵 $\boldsymbol{G}_n = \boldsymbol{G}(\varphi_0, \varphi_1, \cdots, \varphi_n)$ 为格拉姆矩阵, 由于 $\varphi_0, \varphi_1, \cdots, \varphi_n$ 线性无关, 故 $\det \boldsymbol{G}_n \neq 0$. 于是方程组(3.5.7)存在惟一解 $a_k = a_k^*(k=0,1,\cdots,n)$, 从而得到

$$r_n^*(x)=a_0^*\varphi_0(x)+a_1^*\varphi_1(x)+\cdots+a_n^*\varphi_n(x).$$

可以证明 $r_n^*(x)$满足式(3.5.5),即对任何 $r_n(x)\in\Phi_n$,有

$$\int_a^b\rho(x)[f(x)-r_n^*(x)]^2\mathrm{d}x\leqslant\int_a^b\rho(x)[f(x)-r_n(x)]^2\mathrm{d}x.$$

若记 $\delta(x)=f(x)-r_n^*(x)$,则可得到

$$\begin{aligned}\|\delta(x)\|_2^2&=(f(x)-r_n^*(x),f(x)-r_n^*(x))\\&=(f(x),f(x))-(f(x),r_n^*(x))\\&=\|f\|_2^2-\sum_{k=0}^n a_k^*(f,\varphi_k),\end{aligned}\tag{3.5.8}$$

称为最佳逼近多项式的**平方误差**,而 $\|\delta(x)\|_2$ 称为**均方误差**.

若 $\rho(x)\equiv1$,并取 $\varphi_k(x)=x^k(k=0,1,\cdots,n)$,则 $r_n^*(x)$就是由式(3.5.3)得到的最佳平方逼近多项式.

若区间取为$[0,1]$,$f(x)\in C[0,1]$,则在$[0,1]$上的最佳平方逼近多项式

$$r_n^*(x)=a_0^*+a_1^*x+\cdots+a_n^*x^n,$$

此时方程(3.5.7)中的系数及自由项为

$$(\varphi_j,\varphi_k)=\int_0^1x^{j+k}\mathrm{d}x=\frac{1}{j+k+1}\quad(j,k=0,1,\cdots,n),$$

$$(f,\varphi_j)=\int_0^1f(x)x^j\mathrm{d}x=d_j\quad(j=0,1,\cdots,n).$$

若用 $\boldsymbol{H}_n$ 表示格拉姆矩阵 $\boldsymbol{G}_n=G(1,x,\cdots,x^n)$,即

$$\boldsymbol{H}_n=\begin{pmatrix}1&1/2&\cdots&1/n+1\\1/2&1/3&\cdots&1/n+2\\\vdots&\vdots&&\vdots\\1/n+1&1/n+2&\cdots&1/2n+1\end{pmatrix},\tag{3.5.9}$$

则称 $\boldsymbol{H}_n$ 为**希尔伯特**(Hilbert)**矩阵**. 记 $\boldsymbol{a}=(a_0,a_1,\cdots,a_n)^{\mathrm{T}}$,$\boldsymbol{d}=(d_0,d_1,\cdots,d_n)^{\mathrm{T}}$,则得法方程

$$\boldsymbol{H}_n\boldsymbol{a}=\boldsymbol{d}.\tag{3.5.10}$$

由于 $\det\boldsymbol{H}_n\neq0$,故方程组有解 $\boldsymbol{a}_k=\boldsymbol{a}_k^*$ $(k=0,1,\cdots,n)$. 但希尔伯

特矩阵是病态矩阵，通常解的误差很大，当 $n\geqslant 3$ 时结果并不可靠，解(3.5.10)的法方程通常只用于 $n\leqslant 2$ 的简单情形，一般情形可用正交多项式作为基. 此时法方程(3.5.7)的系数矩阵是对角阵. 具体算法将在 3.6 节介绍.

3.6 用正交函数族作最佳平方逼近

3.6.1 最佳平方逼近与广义傅里叶级数

从上节的讨论可知求函数的最佳平方逼近广义多项式可归结为求法方程(3.5.7)的解，若以单项式 $\{1,x,\cdots,x^n\}$ 为基求最佳平方逼近多项式，其法方程是病态的，如果基 $\{\varphi_0(x),\cdots,\varphi_n(x)\}$ 是正交函数族，则方程组(3.5.7)的系数矩阵 $\boldsymbol{G}_n=G(\varphi_0,\varphi_1,\cdots,\varphi_n)$ 为非奇异对角阵，且方程组(3.5.7)的解为

$$a_k^* = (f,\varphi_k)/(\varphi_k,\varphi_k) \quad (k=0,1,\cdots,n), \tag{3.6.1}$$

于是 $f(x)\in C[a,b]$ 的最佳平方逼近广义多项式为

$$r_n^*(x) = \sum_{k=0}^{n} \frac{(f,\varphi_k)}{\|\varphi_k\|_2^2}\varphi_k(x). \tag{3.6.2}$$

由(3.5.8)可得平方误差

$$\begin{aligned}\|\delta_n(x)\|_2^2 &= \|f(x)-r_n^*(x)\|_2^2 \\ &= \|f\|_2^2-\sum_{k=0}^{n}\frac{(f,\varphi_k)^2}{\|\varphi_k\|_2^2}\geqslant 0.\end{aligned} \tag{3.6.3}$$

若 $f(x)\in C[a,b]$ 按正交函数族 $\{\varphi_k(x)\}_{k=0}^{\infty}$ 展开，即由式(3.6.1)计算 a_k^* $(k=0,1,\cdots)$ 得到级数

$$\sum_{k=0}^{\infty}\left[\frac{(f,\varphi_k)}{\|\varphi_k\|^2}\right]\varphi_k(x), \tag{3.6.4}$$

称为函数 $f(x)$ 的**广义傅里叶(Fourier)级数**，系数 a_k^* $(k=0,1,\cdots)$ 由式(3.6.1)给出，称为广义傅里叶系数，它是傅里叶级数的直接推广. 同样，由式(3.6.3)，当 $n\to\infty$ 时，有

$$\sum_{k=0}^{\infty}\left[\frac{(f,\varphi_k)}{\|\varphi_k\|_2^2}\right]^2 \leqslant \|f\|_2^2, \qquad (3.6.5)$$

称为**广义贝塞尔不等式**.

若函数族$\{\varphi_0,\varphi_1,\cdots,\varphi_n,\cdots\}$为正交多项式,则$r_n^*(x)$也是$n$次多项式,它就是$f(x)$的最佳平方逼近多项式,并有以下收敛性结果,即

$$\lim_{n\to\infty}\|f(x)-r_n^*(x)\|_2 = 0. \qquad (3.6.6)$$

3.6.2 用勒让德多项式作平方逼近

在有限区间$[a,b]$上求函数$f(x)\in C[a,b]$的最佳平方逼近,当$\rho(x)\equiv 1$时,只要将$f(\cdot)$按勒让德多项式展开即可,当$[a,b]\neq[-1,1]$时,应作变换.令

$$x=\frac{b-a}{2}t+\frac{b+a}{2},$$

则$-1\leqslant t\leqslant 1$,于是$f(x)$可变换为

$$F(t)=f\left(\frac{b-a}{2}t+\frac{b+a}{2}\right) \quad (-1\leqslant t\leqslant 1).$$

对$F(t)$用$[-1,1]$上的勒让德多项式展开部分和即得$f(x)$在$[a,b]$上的最佳平方逼近,因此,只要考虑$f(x)\in C[-1,1]$,按勒让德多项式$\{\mathrm{P}_0,\mathrm{P}_1,\cdots,\mathrm{P}_n,\cdots\}$展开,由式(3.6.2)得

$$f(x)\sim\sum_{k=0}^{\infty}c_k^*\mathrm{P}_k(x),$$

其中c_k^*可利用式(3.6.1),将φ_k改为P_k,a_k^*改为c_k^*即得

$$c_k^*=\frac{(f,\mathrm{P}_k)}{(\mathrm{P}_k,\mathrm{P}_k)}=\frac{2k+1}{2}(f,\mathrm{P}_k).$$

于是f在$[-1,1]$上的最佳平方逼近记为

$$S_n^*(x)=\sum_{k=0}^{n}\frac{2k+1}{2}(f,\mathrm{P}_k)\mathrm{P}_k(x), \qquad (3.6.7)$$

这里

$$P_k(x) = \frac{1}{2^k k!}\frac{d^k}{dx^k}[(x^2-1)^k] \quad (k=1,2,\cdots,n),$$

$$P_0(x) = 1.$$

由式(3.6.3)可得到

$$\| f - S_n^* \|_2^2 = \| f \|_2^2 - \sum_{k=0}^{n} \frac{2k+1}{2}[(f,P_k)]^2.$$

用式(3.6.6)求 f 的最佳平方逼近不用求解方程组(3.5.7),从而避免了解法方程出现的病态问题.

由式(3.6.6)可直接得到平均收敛,即

$$\lim_{n\to\infty} \| f(x) - S_n^*(x) \|_2 = 0,$$

如果 $f \in C^2[-1,1]$还可得到一致收敛的结论.

定理 3.6.1 设 $f(x) \in C^2[-1,1]$,$S_n^*(x)$由式(3.6.7)给出,则对任意 $x \in [-1,1]$和 $\varepsilon > 0$,当 n 充分大有

$$| f(x) - S_n^*(x) | \leqslant \frac{\varepsilon}{\sqrt{n}}.$$

例 3.6.2 求 $f(x) = e^x$ 在$[-1,1]$上三次最佳平方逼近.

解 由式(3.6.7)计算(f,P_n) $(n=0,1,2,3)$:

$$\begin{cases} (f,P_0) = \int_{-1}^{1} e^x dx \approx 2.350388, \\ (f,P_1) = \int_{-1}^{1} x e^x dx \approx 0.735759, \\ (f,P_2) = \int_{-1}^{1} \left(\frac{3}{2}x^2 - \frac{1}{2}\right) e^x dx \approx 0.143124, \\ (f,P_3) = \int_{-1}^{1} \left(\frac{5}{2}x^3 - \frac{3}{2}x\right) e^x dx \approx 0.201302. \end{cases}$$

由此可得

$$S_3^*(x) = 1.145194P_0(x) + 1.103639P_1(x) + 0.357810P_2(x) + 0.704557P_3(x),$$

即

$$S_3^*(x) = 0.996294 + 0.997955x + 0.536722x^2 + 0.176139x^3,$$

$$\| e^x - S_3^*(x) \|_2^2 \leqslant 0.0084,$$

$$\| e^x - S_3^*(x) \|_\infty \leqslant 0.0112,$$

这里得到的 $S_3^*(x)$就是式(3.5.4)的结果.

3.6.3 截断切比雪夫级数

如果 $f \in C[a,b]$，当正交多项式取切比雪夫多项式$\{T_k(x)\}$时，广义傅里叶级数

$$\sum_{k=0}^{\infty} c_k^* T_k(x) \tag{3.6.8}$$

称为函数 f 在$[-1,1]$上的切比雪夫级数，其中系数由公式(3.6.2)得到

$$\begin{cases} c_0^* = \dfrac{(T_0,f)}{(T_0,T_0)} = \dfrac{1}{\pi}(T_0,f) = \dfrac{1}{\pi}\displaystyle\int_{-1}^{1} \dfrac{f(x)}{\sqrt{1-x^2}}\mathrm{d}x, \\ c_k^* = \dfrac{(T_k,f)}{(T_k,T_k)} = \dfrac{2}{\pi}(T_k,f) = \dfrac{2}{\pi}\displaystyle\int_{-1}^{1} \dfrac{T_k(x)f(x)}{\sqrt{1-x^2}}\mathrm{d}x, \end{cases}$$

$$k = 1,2,\cdots.$$

$T_k(x) = \cos(k\arccos x)$为切比雪夫多项式.

若令 $x = \cos\theta, 0 \leqslant \theta \leqslant \pi$，则式(3.6.8)就是 $f(\cos\theta)$的傅里叶级数，其中

$$c_0^* = \frac{1}{\pi}\int_0^{\pi} f(\cos\theta)\mathrm{d}\theta,$$

$$c_k^* = \frac{2}{\pi}\int_0^{\pi} f(\cos\theta)\cos k\theta\mathrm{d}\theta \quad (k = 1,2,\cdots).$$

根据傅里叶级数理论可知，如果 $f''(x)$在区间$[-1,1]$上分段连续，则切比雪夫级数(3.6.8)在区间$[-1,1]$上一致收敛于$f(x)$，即

$$f(x) = \sum_{k=0}^{\infty} c_k^* T_k(x).$$

取它的前 n 项部分和

$$c_n^*(x) = \sum_{k=0}^{n} c_k^* T_k(x), \tag{3.6.9}$$

则

$$f(x) - c_n^*(x) \approx c_{n+1}^* T_{n+1}(x).$$

由于 $T_{n+1}(x)$有 $n+2$ 个轮流为"+"、"−"的偏差点，所以 $f(x)-c_n^*(x)$近似地有 $n+2$ 个偏差点，故由切比雪夫定理 3.2.4，$c_n^*(x)$可作为 $f(x)$的近似最佳一致逼近多项式. 事实上，许多函数的截断切比雪夫级数(3.6.9)都很近似最佳一致逼近多项式，对具有高阶导数的函数 f，其误差 $\| f-c_n^* \|_\infty$ 一般不超过最小偏差 E_n 的百分之五. 最佳一致逼近多项式 $p_n^*(x)$难于计算，因此，通常都用 $c_n^*(x)$作为近似最佳一致逼近多项式.

例 3.6.3 求 $f(x)=e^x$ 在区间$[-1,1]$上的截断切比雪夫级数.

解 由公式(3.6.9)，系数为

$$c_0^* = \frac{1}{\pi}\int_{-1}^{1} \frac{e^x}{\sqrt{1-x^2}}dx = \frac{1}{\pi}\int_0^\pi e^{\cos\theta}d\theta,$$

$$c_k^* = \frac{2}{\pi}\int_{-1}^{1} f(x)\frac{T_k(x)}{\sqrt{1-x^2}}dx$$

$$= \frac{2}{\pi}\int_0^\pi e^{\cos\theta}\cos(k\theta)d\theta \quad (k=1,2,\cdots).$$

用数值积分方法可算出

$$c_0^* = 1.26606588, \quad c_1^* = 1.13031821,$$

$$c_2^* = 0.27149534, \quad c_3^* = 0.04433685,$$

$$c_4^* = 0.00547424, \quad c_5^* = 0.00054293.$$

由式(3.6.9)及 $T_k(x)$的表达式，可得

$$c_1^*(x) = 1.266 + 1.130x,$$

$$c_3^*(x) = 0.994571 + 0.997308x + 0.542991x^2 + 0.177347x^3.$$

用 $c_1^*(x)$ 及 $c_3^*(x)$ 逼近 e^x 的误差为

$$\|e^x - c_1^*(x)\|_\infty \approx 0.32,$$

$$\|e^x - c_3^*(x)\|_\infty \approx 0.00607.$$

它与 e^x 在 $[-1,1]$ 上的三次最佳一致逼近偏差 0.00553 相差不多. 图 3.8 给出了误差函数 $y = e^x - c_3^*(x)$ 的图形. 它与图 3.4 给出的最佳逼近误差分布很相似. 这说明用截断的切比雪夫级数 $c_n^*(x)$, 可作为最佳一致逼近的很好近似.

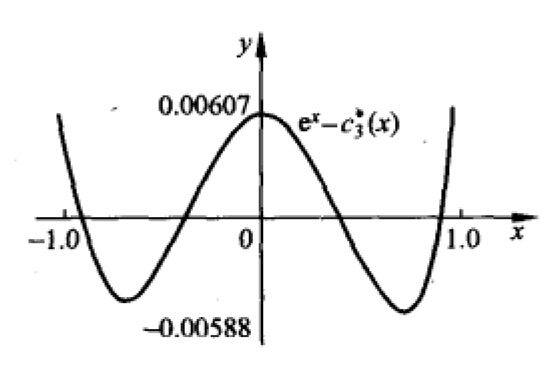


图 3.8

3.7 近似最佳一致逼近

由于函数的最佳一致逼近多项式计算困难, 因此, 实际计算往往只求近似的最佳一致逼近, 主要工具是利用切比雪夫多项式的良好性质, 其中以式(3.6.9)给出的截断切比雪夫级数 $c_n^*(x)$ 为最好的近似算法, 另外, 还有下面两种近似方法.

3.7.1 泰勒级数项数的节约

由例 3.2.2 可知, 函数泰勒展开容易计算, 但误差分布极不均匀. 为了改善函数逼近的误差分布, 可利用 $T_n(x)$ 均匀分布的性质, 对泰勒部分和 $p_n(x)$ 进行改造, 设

$$\| f(x) - p_n(x) \|_\infty \leqslant \varepsilon_n \ll \varepsilon, \tag{3.7.1}$$

其中 $p_n(x)$ 由(3.2.2)给出，$\varepsilon>0$ 是要求逼近的精度. 由于

$$\widetilde{T}_n(x) = \frac{1}{2^{n-1}} T_n(x) = x^n + Q(x), \tag{3.7.2}$$

其中 $Q(x)$ 为次数小于 n 的多项式. 现设

$$p_n(x) = a_0 + a_1 x + \cdots + a_n x^n,$$

由式(3.7.2)可得

$$p_n(x) = a_0 + a_1 x + \cdots + a_{n-1} x^{n-1} + a_n \left[-Q(x) + \frac{1}{2^{n-1}} T_n(x) \right].$$

令

$$M_{n,n-1}(x) = a_0 + a_1 x + \cdots + a_{n-1} x^{n-1} - a_n Q(x),$$

$$\varepsilon_{n-1} = \left| \frac{a_n}{2^{n-1}} \right|,$$

若

$$\varepsilon_n + \varepsilon_{n-1} \leqslant \varepsilon$$

则

$$\begin{aligned} \| f(x) - M_{n,n-1} \|_\infty &\leqslant \| f - p_n \|_\infty + \| p_n - M_{n,n-1} \| \\ &\leqslant \varepsilon_n + \varepsilon_{n-1} \leqslant \varepsilon. \end{aligned}$$

这时用 $M_{n,n-1}(x)$ 作为 $f(x)$ 新的逼近多项式，次数已降低一次，再由 $M_{n,n-1}(x)$ 出发，用同样方法. 若 $\varepsilon_n + \varepsilon_{n-1} + \varepsilon_{n-2} \leqslant \varepsilon$，则 $M_{n,n-1}(x)$ 可再降低一次，一直做到次数不能再降低为止.

此时得到的逼近多项式记作 $M_{n,m}(x)(m<n)$.

例 3.7.1 $f(x)=e^x, |x| \leqslant 1$，其五阶泰勒展开为

$$p_5(x) = 1 + x + \frac{1}{2}x^2 + \frac{1}{6}x^3 + \frac{1}{24}x^4 + \frac{1}{120}x^5,$$

$$\| f - p_5 \|_\infty \leqslant \frac{1}{6!} e \approx 0.00377.$$

由于

$$x^5=\frac{1}{16}T_5(x)+\frac{5}{4}x^3-\frac{5}{16}x,$$

$$x^4=\frac{1}{8}T_4(x)+x^2-\frac{1}{8},$$

$$p_5(x)=M_{5,3}(x)+\frac{1}{192}T_4(x)+\frac{1}{1920}T_5(x),$$

其中

$$M_{5,3}(x)=0.994792+0.997396x+0.541667x^2+0.177083x^3,$$

$$\|e^x-M_{5,3}\|_\infty\leqslant 0.00377+\frac{1}{192}+\frac{1}{1920}\approx 0.0095,$$

进一步若由 $p_6(x)$出发,可求得

$$M_{6,3}(x)=0.994575+0.997396x+0.542969x^2+0.177083x^3, \tag{3.7.3}$$

$$\|e^x-M_{6,3}\|_\infty\leqslant 0.00651.$$

此结果已接近 $c_3^*(x)$结果,如从 $p_7(x)$或 $p_8(x)$出发求 $M_{7,3}(x)$或 $M_{8,3}(x)$,其结果将会更接近 $p_3^*(x)$,故可作为求最佳一致逼近的近似算法.

3.7.2 切比雪夫多项式零点插值

利用插值多项式 $L_n(x)$逼近 $f(x)$,若插值点选择合适,可使误差减小.设 $f\in C[-1,1]$,选 $T_n(x)$的零点 $x_k=\cos\frac{2k-1}{2n}\pi(k=1,\cdots,n)$为插值点,构造插值多项式 $L_{n-1}(x)$,由于

$$\omega_n(x)=(x-x_1)\cdots(x-x_n)=\frac{1}{2^{n-1}}T_n(x),$$

故插值误差

$$\|f-L_{n-1}\|_\infty\leqslant\frac{M_n}{n!}\max_{-1\leqslant x\leqslant 1}|\omega_n(x)|=\frac{M_n}{n!}\frac{1}{2^{n-1}},$$

其中

$$M_n = \max_{-1\leqslant x\leqslant 1} |f^{(n)}(x)|.$$

由切比雪夫多项式性质(2)可知，$\frac{1}{2^{n-1}}T_n(x)$在$[-1,1]$上最高项系数为1的n次多项式中与零偏差最小，即$\max\limits_{-1\leqslant x\leqslant 1}|\omega_n(x)|=\min$，这说明选$T_n(x)$的零点做插值点在固定$M_n$后其误差最小，因此，可利用这种插值多项式$L_{n-1}(x)$作为近似最佳一致逼近. 当$f(x)\in C[a,b]$时，只要通过变换

$$x=\frac{b-a}{2}t+\frac{b+a}{2}$$

即可在$t\in[-1,1]$上应用$T_n(t)$的零点做插值点，求得所要求的插值多项式.

例 3.7.2 求$f(x)=e^x$在$[-1,1]$上的插值多项式.

解 在$[-1,1]$求$L_3(x)$，用$T_4(x)$的零点$x_k=\cos\frac{2k-1}{8}\pi$ $(k=1,2,3,4)$做插值点.

$$x_1=\cos\frac{1}{8}\pi\approx 0.9238795,\quad x_2=\cos\frac{3}{8}\pi\approx 0.3826834,$$

$$x_3=\cos\frac{5}{8}\pi\approx -0.3826834,\quad x_4=\cos\frac{7}{8}\pi\approx 0.9238795.$$

构造插值多项式可得

$$L_3(x)=0.994584+0.998967x+0.542900x^2+0.175176x^3, \tag{3.7.4}$$

$$\|e^x-L_3\|_\infty\leqslant 0.00666.$$

前面已给出函数$f(x)\in C[a,b]$求逼近多项式的各种方法，下面以$f(x)=e^x$在$[-1,1]$上三次逼近多项式为例，将已算出的各种逼近多项式的误差列在表3.1中，不同多项式的误差均用$\|f-p\|_\infty$表示.

表 3.1

逼近方法	误差 $\|f-p\|_\infty$
泰勒多项式 $p_3(x)$	0.0516
级数节约项数 $M_{5,3}(x)$	0.0095
级数节约项数 $M_{6,3}(x)$	0.00651
勒让德最小平方逼近 $S_3^*(x)$	0.0112
切比雪夫最小平方逼近 $C_3^*(x)$	0.00607
切比雪夫零点插值 $L_3(x)$	0.00666
最佳一致逼近 $p_3^*(x)$	0.00553

从表中可看到,利用切比雪夫多项式得到的一些逼近多项式已经是最佳一致逼近的较好近似,它们计算较简单,而列梅兹算法计算复杂,通常不大使用.

3.8 曲线拟合的最小二乘法

3.8.1 基本原理

在科学实验或统计方法研究中,往往要从一组实验数据(x_i,y_i) $(i=1,\cdots,m)$中寻找自变量 x 和因变量 y 之间的一个函数关系式 $y=f(x)$,从图形上看就是由给定的 m 个点求曲线拟合的问题.由于科学实验得到的数据往往带有观测误差,如果要求曲线 $y=f(x)$必须通过给定的点(x_i,y_i),不但会把观测误差保留下来,而且 $y=f(x)$的曲线也不一定表示实验数据的客观规律.因此,对这类问题不能使用已介绍的插值方法,而应采用最小二乘法,所谓曲线拟合的最小二乘法,就是由给定数据$(x_i,y_i)(i=1,\cdots,m)$,确定拟合曲线类型 $y=P(x,a_0,\cdots,a_n)$,然后根据在给定点误差平方和

$$\sum_{i=1}^{m}[P(x_i,a_0,\cdots,a_n)-y_i]^2 \tag{3.8.1}$$

最小的原则定出参数 $a_k(k=0,\cdots,n)$，从而得到所求的拟合曲线方程 $y=P^*(x)$.

例 3.8.1 已给一组实验数据如下：

i	1	2	3	4
x_i	2	4	6	8
y_i	2	11	28	40

将它标在坐标纸上，如图 3.9 所示. 容易看出这些点在一条直线附近，因此，可以选

$$P(x)=a_0+a_1x.$$

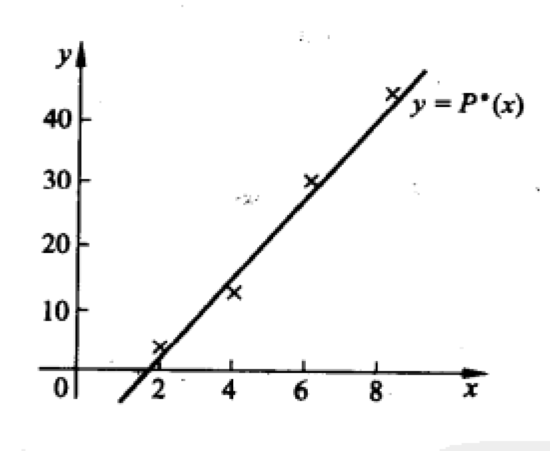


图 3.9

由式(3.8.1)，令

$$F(a_0,a_1)=\sum_{i=1}^{4}[(a_0+a_1x_i)-y_i]^2,$$

为求二元函数 F 的最小点，由极值必要条件可得

$$\begin{cases}\dfrac{\partial F}{\partial a_0}=\displaystyle\sum_{i=1}^{4}2(a_0+a_1x_i-y_i)=0,\\[2ex]\dfrac{\partial F}{\partial a_1}=\displaystyle\sum_{i=1}^{4}2x_i(a_0+a_1x_i-y_i)=0.\end{cases}$$

化简，得

$$
\begin{cases}
4a_0 + (\sum_{i=1}^{4} x_i)a_1 = \sum_{i=1}^{4} y_i, \\
(\sum_{i=1}^{4} x_i)a_1 + (\sum_{i=1}^{4} x_i^2)a_1 = \sum_{i=1}^{4} x_i y_i.
\end{cases}
$$

将已知数据代入后得

$$
\begin{cases}
4a_0 + 20a_1 = 81, \\
20a_0 + 120a_1 = 536,
\end{cases}
$$

解得

$$
a_0 = -12.5, \quad a_1 = 6.55.
$$

于是得到拟合曲线方程

$$
y = P^*(x) = 6.55x - 12.5.
$$

3.8.2 线性最小二乘逼近

设给定数据 (x_i, y_i) $(i=1,\cdots,m)$，假定拟合曲线方程为 $y=P(x,a_0,\cdots,a_n)$，令

$$
F(a_0,\cdots,a_n) = \sum_{i=1}^{m} \omega_i [P(x_i,a_0,\cdots,a_n) - y_i]^2 \quad (n < m+1), \tag{3.8.2}
$$

$\omega_i > 0 (i=1,\cdots,m)$ 称为点 (x_i, y_i) 的权系数，最小二乘法就是求参量 $a_i (i=0,1,\cdots,n)$，使 $F(a_0,\cdots,a_n)$ 最小，即求 $a_i = a_i^*$ 使

$$
F(a_0^*, a_1^*, \cdots, a_n^*) \leqslant F(a_0,\cdots,a_n).
$$

由多元函数极值必要条件可得

$$
\frac{\partial F}{\partial a_k} = 0 \quad (k = 0,1,\cdots,n). \tag{3.8.3}
$$

当 F 为 a_i 的非线性函数时称为非线性最小二乘问题，这时式(3.8.3)为非线性方程组. 如果 F 为 a_i 的线性函数，则称为线性最小二乘问题，此时可令

$$P(x) = a_0\varphi_0(x) + a_1\varphi_1(x) + \cdots + a_n\varphi_n(x),$$

其中 $\varphi_0, \varphi_1, \cdots, \varphi_n$ 是线性无关的. 由式(3.8.2)得

$$F(a_0, \cdots, a_n) = \sum_{i=0}^{m}\omega_i\Big[\sum_{k=0}^{m}a_k\varphi_k(x_i) - y_i\Big]^2, \quad (3.8.4)$$

由式(3.8.3)可得

$$\begin{aligned}\frac{\partial F}{\partial a_k} &= 2\sum_{i=1}^{m}\omega_i\Big[\sum_{k=0}^{n}a_k\varphi_k(x_i) - y_i\Big]\varphi_k(x_i)\\ &= 0 \quad (k = 0,1,\cdots,n).\end{aligned} \quad (3.8.5)$$

若记 $y_i = f(x_i)$,

$$(\varphi_k, \varphi_i) = \sum_{i=1}^{m}\omega_i\varphi_k(x_i)\varphi_j(x_i),$$

$$(f, \varphi_j) = \sum_{i=1}^{m}\omega_i f(x_i)\varphi_j(x_i),$$

则式(3.8.5)可改写为

$$\sum_{k=0}^{n}(\varphi_k, \varphi_j)a_k = (f, \varphi_j) \quad (j = 0,1,\cdots,n), \quad (3.8.6)$$

这个方程组也称**法方程**. 它是关于 $a_0, \cdots, a_n$ 的 $n+1$ 阶线性方程组,其系数矩阵为

$$\boldsymbol{\Phi} = \begin{bmatrix}(\varphi_0, \varphi_0) & (\varphi_0, \varphi_1) & \cdots & (\varphi_0, \varphi_n)\\ \vdots & \vdots & & \vdots\\ (\varphi_n, \varphi_0) & (\varphi_n, \varphi_1) & \cdots & (\varphi_n, \varphi_n)\end{bmatrix}.$$

为了证明法方程(3.8.6)的解存在惟一,对函数族$\{\varphi_0(x), \varphi_1(x), \cdots, \varphi_n(x)\}$还要求满足哈尔条件.

定义 3.8.2 如果函数族$\{\varphi_0(x), \varphi_1(x), \cdots, \varphi_n(x)\}$在有限点集 $X = \{x_1, x_2, \cdots, x_m\}$中的任意 $n+1(n \leqslant m+1)$个点$\{x_{j_l}, l = 0, 1, \cdots, n\}$上都有

$$\det\begin{bmatrix}\varphi_0(x_{j_0}) & \varphi_1(x_{j_0}) & \cdots & \varphi_n(x_{j_0})\\ \vdots & \vdots & & \vdots\\ \varphi_0(x_{j_n}) & \varphi_1(x_{j_n}) & \cdots & \varphi_n(x_{j_n})\end{bmatrix}\neq 0$$

则称函数族$\{\varphi_i(x),i=0,1,\cdots,n\}$在点集 X 上满足哈尔条件.

这个定义实际上等价于：函数族$\{\varphi_i(x),i=0,1,\cdots,n\}$的任意线性组合在点集 X 上至多只有 n 个不同零点.

显然单项式 $1,x,\cdots,x^n$ 在任意 $m+1$ 个点的点集 X 上满足 Haar 条件.

可以证明，如果 $\Phi_n=\text{span}\{\varphi_0,\varphi_1,\cdots,\varphi_n\}$在$\{x_i\}_1^m$ 上满足哈尔条件，则法方程(3.8.6)的系数矩阵 $\boldsymbol{\Phi}$ 非奇异，即 $\det\boldsymbol{\Phi}\neq 0$. 于是方程组(3.8.6)存在惟一的解 $a_k=a_k^*\quad(k=0,1,\cdots,n)$，从而得到离散数据$(x_i,y_i)\quad(i=0,1,\cdots,n)$的最小二乘拟合曲线

$$y=P_n^*(x)=a_0^*\varphi_0(x)+a_1^*\varphi_1(x)+\cdots+a_n^*\varphi_n(x).\tag{3.8.7}$$

可以证明，这样得到的 $P_n^*(x)$确实使函数(3.8.4)取得极小，故 $P_n^*(x)$即为所求的最小二乘解. 若取 $\varphi_k(x)=x^k\quad(k=0,1,\cdots,n)$，这时得到的最小二乘解为

$$P_n^*(x)=a_0^*+a_1^*x+\cdots+a_n^*x^n,$$

与连续情形最小平方逼近相似，相应的系数矩阵 $\boldsymbol{\Phi}$ 也是病态的. 因此，当 $n\geqslant 3$ 时，用这种方法计算误差较大，通常要用正交化方法计算，见 3.8 节. 对于实际问题中遇到的大量属于 $n\leqslant 2$ 的情形，均可用本节介绍的算法.

例 3.8.3 在某化学反应里，根据实验，生成物的浓度与时间关系如下表，求浓度 y 与时间 t 的拟合曲线.

时间 t/min	1	2	3	4	5	6	7	8
浓度 $y\times 10^{-3}$	4.00	6.40	8.00	8.80	9.22	9.50	9.70	9.80
时间 t/min	9	10	11	12	13	14	15	16
浓度 $y\times 10^{-3}$	10.0	10.20	10.32	10.42	10.50	10.55	10.59	10.60

解 第一步先将实验数据表点在坐标纸上，见图 3.10.

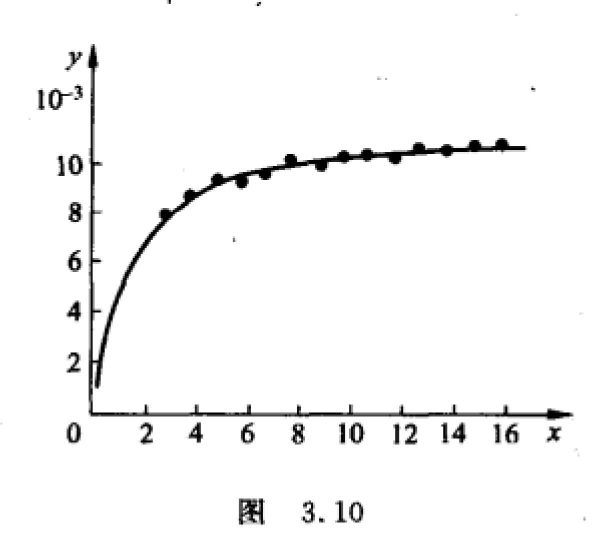


图 3.10

第二步，确定经验公式的函数类型. 从图可以看到浓度 y 开始增加快而后逐渐减弱，到一定时间后基本稳定，$t\to\infty$ 时 $y\to$ 常数，有一水平渐近线；反应未开始时 $t=0,y=0$. 根据这些分析可设想下列两种曲线：

双曲线型：

$$\frac{1}{y}=a+\frac{b}{t},$$

即 $y=\dfrac{t}{at+b}$，此时 $t=0,y=0$；$t\to\infty,y\to\dfrac{1}{a}$.

指数型：

$$y=ae^{b/t},$$

当 $b<0$ 时，有 $t\to0\quad y\to0$；$\quad t\to+\infty,\quad y\to a$.

第三步，求最小二乘法的解. 为确定系数 a,b，可以分别用两种类型计算. 对双曲线型，可作变换 $\bar{y}=\dfrac{1}{y},x=\dfrac{1}{t}$，则得线性模型

$$\bar{y}=a+bx=p(x).$$

拟合曲线数据由 (t_i,y_i) 变换为 $(x_i,\bar{y}_i)$ $(i=1,\cdots,16)$. 以下计算与

例 3.8.1 相似,其法方程为

$$\begin{cases} 16a + 3.38073b = 1.8372 \times 10^3, \\ 3.38073a + 1.58435b = 0.52886 \times 10^3, \end{cases}$$

解得

$$a = 80.6621, \quad b = 161.6822,$$

从而得到

$$y = \frac{t}{80.6621t + 161.6822} = F^{(1)}(t). \tag{3.8.8}$$

对指数型

$$y = a e^{b/c}, \tag{3.8.9}$$

两边取对数,得

$$\ln y = \ln a + \frac{b}{t},$$

作变换

$$\hat{y} = \ln y, A = \ln a, x = \frac{1}{t},$$

由(t_i, y_i)计算$(x_i, \hat{y}_i)$,数据$(x_i, \hat{y}_i)$的拟合曲线为线性模型

$$\hat{y} = A + bx = p(x),$$

用线性拟合最小二乘法可求得

$$A = -4.48072, \quad b = -1.0567,$$

$$a = e^A = 11.3253 \times 10^{-3}.$$

由式(3.8.9)可得

$$y = 11.3253 \times 10^{-3} e^{-\frac{1.0567}{t}} = F^{(2)}(t), \tag{3.8.10}$$

第四步,根据求得的函数 $F^{(1)}$ 及 $F^{(2)}$ 计算,得

$$\delta_i^{(1)} = y_i - F^{(1)}(t_i),$$

$$\delta_i^{(2)} = y_i - F^{(2)}(t_i), \quad i = 1, \cdots, 16,$$

$$\max_i | \delta_i^{(1)} | = 0.568 \times 10^{-3},$$

$$\max_i | \delta_i^{(2)} | = 0.277 \times 10^{-3}.$$

可见，$\|\delta^{(2)}\|_\infty$比$\|\delta^{(1)}\|_\infty$小.因此，选择$y=F^{(2)}(t)$做拟合曲线较好.

在用最小二乘法作曲线拟合时，数学模型选择要通过实际计算才能确定，目前有很多计算机配有自动选择数学模型的软件，它通过大量计算可选得误差较小的模型.另外，只要通过变换能化为线性模型的问题均可用本节的算法计算.

3.8.3 用正交多项式作最小二乘拟合

用多项式作最小二乘拟合时，其法方程是病态的，为了避免求病态方程，通常采用正交化方法，对给定点集$\{x_i\}(i=1,\cdots,m)$及权系数$\{\omega_i\}(i=1,\cdots,m)$，如果函数系$\varphi_0(x),\cdots,\varphi_n(x)$满足

$$(\varphi_j,\varphi_k)=\sum_{i=1}^m \omega_i\varphi_j(x_i)\varphi_k(x_i)=\begin{cases}0, & j\neq k,\\ A_k>0, & j=k,\end{cases} \tag{3.8.11}$$

则称$\{\varphi_j\}$关于点集$\{x_i\}$带权$\{\omega_i\}$正交.此时方程(3.8.6)的解为

$$a_k=a_k^*=\frac{(f,\varphi_k)}{(\varphi_k,\varphi_k)}=\frac{\sum_{i=1}^m \omega_i f(x_i)\varphi_k(x_i)}{\sum_{i=1}^m \omega_i\varphi_k^2(x_i)},\quad k=0,1,\cdots,n, \tag{3.8.12}$$

且平方误差为

$$\|\delta\|_2^2=\|f\|_2^2-\sum_{k=0}^n A_k(a_k^*)^2. \tag{3.8.13}$$

若已给数据(x_i,f_i)及权$\omega_i(i=1,\cdots,m)$，可以构造出带权$\{\omega_i\}$正交的多项式$\{p_k(x)\}$，用递推关系表示

$$\begin{cases}p_0(x)=1,\\ p_1(x)=(x-\alpha_1)p_0(x),\\ p_{k+1}(x)=(x-\alpha_{k+1})p_k(x)-\beta_k p_{k-1}(x),\\ \qquad k=1,2,\cdots,n-1,\end{cases} \tag{3.8.14}$$

这里 p_k 是首项系数为 1 的 k 次多项式，待定系数 α_k 及 β_k 可根据 $\{p_k(x)\}$ 的正交性，即

$$(p_k, p_j) = \sum_{i=1}^{m} \omega_i p_k(x_i) p_j(x_i) = \begin{cases} 0, & j \neq k, \\ A_k > 0, & j = k, \end{cases}$$

$$j, k = 0, 1, \cdots, n,$$

求得

$$\begin{cases} \alpha_{k+1} = \dfrac{\sum\limits_{i=1}^{m} \omega_i x_i p_k^2(x_i)}{\sum\limits_{i=1}^{m} \omega_i p_k^2(x_i)} = \dfrac{(x_i p_k, p_k)}{(p_k, p_k)}, \\ \beta_k = \dfrac{\sum\limits_{i=1}^{m} \omega_i p_k^2(x_i)}{\sum\limits_{i=1}^{m} \omega_i p_{k-1}^2(x_i)} = \dfrac{(p_k, p_k)}{(p_{k-1}, p_{k-1})}, \end{cases} \tag{3.8.15}$$

$$k = 0, 1, \cdots, n-1.$$

用点集 $\{x_i\}$ 带权 $\{\omega_i\}$ 正交的多项式 $\{p_k(x)\}$ 为基的广义多项式

$$S_n(x) = a_0 p_0(x) + a_1 p_1(x) + \cdots + a_n p_n(x)$$

做最小二乘法，由式(3.8.12)可得

$$a_k = a_k^* = \frac{(f, p_k)}{(p_k, p_k)} = \frac{\sum\limits_{i=1}^{m} \omega_i f_i p_k(x_i)}{\sum\limits_{i=1}^{m} \omega_i p_k^2(x_i)}, \tag{3.8.16}$$

于是

$$S_n^*(x) = a_0^* p_0(x) + a_1^* p_1(x) + \cdots + a_n^* p_n(x)$$

就是所求的最小二乘拟合曲线，其平方误差由式(3.8.13)给出. 用正交化方法求 $S_n^*(x)$ 时，只要对 $k = 0, 1, \cdots, n-1$ 同时用公式(3.8.14)、(3.8.15)及(3.8.16)计算 $p_k(x)$ 及 a_k^*，其计算方法简单，又不用解方程组，因此是一个较好的算法，一般数学库中有用这个算法编制的标准程序供用户调用.

3.8.4 多元最小二乘拟合

最小二乘法的有关概念与结论，可以推广到多元函数. 假如，假定已知一组正数 $\omega_i(i=1,\cdots,m)$ 和多元函数

$$y = f(x_1,\cdots,x_l)$$

的一组测量数据 $(x_{1i},\cdots,x_{li},y_i)(i=1,\cdots,m)$，要求函数

$$p(x_1,\cdots,x_l) = \sum_{k=0}^{n} a_k\varphi_k(x_1,\cdots,x_l) \quad (n < m+1),$$

使得

$$F(a_0,\cdots,a_n) = \sum_{i=1}^{m}\omega_i[y_i - p(x_{1i},\cdots,x_{li})]^2$$

最小，这也是线性最小二乘拟合问题.

$\{\omega_i\}$ 为权系数，其中系数 $a_0,a_1,\cdots,a_n$ 同样满足法方程组(3.8.6)，只是这里

$$(\varphi_j,\varphi_k) = \sum_{i=1}^{m}\omega_i\varphi_j(x_{1i},\cdots,x_{li})\varphi_k(x_{1i},\cdots,x_{li}).$$

求解法方程组得到的 $y=p(x_1,\cdots,x_l)$，称为多元函数 $y=f(x_1,\cdots,x_l)$ 的最小二乘拟合函数.

3.9 傅里叶逼近

当 $f(x)$ 是定义在 $-\infty<x<\infty$ 的周期函数时，用三角函数逼近 $f(x)$ 比用多项式逼近更为合适. 用三角函数作最小平方逼近，亦即函数的傅里叶逼近及**快速傅里叶变换**(fast fourier transform，FFT).

3.9.1 最佳平方逼近与三角插值

设 $f(x)$ 是以 2π 为周期的平方可积函数，用**三角多项式**(trigonometric polynomials)

$$S_n(x) = a_0 + a_1\cos x + b_1\sin x + \cdots + a_n\cos nx + b_n\sin nx$$

做逼近函数，由例 3.4.4 可知三角函数族

$$1, \cos x, \sin x, \cdots, \cos kx, \sin kx, \cdots$$

是正交函数族，于是 $f(x)$ 在 $[0,2\pi]$ 上的最佳平方三角逼近是

$$S_n^*(x) = \frac{1}{2}a_0^* + \sum_{k=1}^{n}(a_k^* \cos kx + b_k^* \sin kx) \tag{3.9.1}$$

其中

$$a_k^* = \frac{1}{\pi}\int_0^{2\pi} f(x)\cos kx \mathrm{d}x \quad (k = 0,1,\cdots,n),$$

$$b_k^* = \frac{1}{\pi}\int_0^{2\pi} f(x)\sin kx \mathrm{d}x \quad (k = 1,2,\cdots,n),$$

称为傅里叶系数. 由 $f(x)$ 的傅里叶系数得到的级数

$$\frac{1}{2}a_0^* + \sum_{k=1}^{\infty}(a_k^* \cos kx + b_k^* \sin kx) \tag{3.9.2}$$

称为 $f(x)$ 的傅里叶级数. 只要 $f'(x)$ 在 $[0,2\pi]$ 上分段连续，则级数(3.9.2)一致收敛于 $f(x)$.

与上述情形对应，当 $f(x)$ 在给定离散点集 $\left\{x_i = \frac{2\pi}{n}i\right\}$ 上的值已知时，可类似定义离散情形的正交性并得到离散傅里叶系数. 对奇数个点的情形，

$$x_i = \frac{2\pi i}{2m+1} \quad (i = 0,1,\cdots,2m), \tag{3.9.3}$$

对 $k, j = 0,1,\cdots,m$ 有

$$\begin{cases} \sum_{i=0}^{2m} \sin jx_i \sin kx_i = \begin{cases} 0, & j \neq k, j = k = 0, \\ \frac{2m+1}{2}, & j = k \neq 0; \end{cases} \\ \sum_{i=0}^{2m} \cos jx_i \cos kx_i = \begin{cases} 0, & j \neq k, \\ \frac{2m+1}{2}, & j = k \neq 0, \\ 2m+1, & j = k = 0; \end{cases} \\ \sum_{i=0}^{2m} \cos jx_i \sin kx_i = 0, \quad \text{对所有 } j,k. \end{cases} \tag{3.9.4}$$

这就证明了三角函数 $\{1, \sin x, \cos x, \cdots, \sin mx, \cos mx\}$ 关于点集(3.9.3)正交.

假定 $f(\cdot)$ 在点集(3.9.3)上的观测值为 $f_i = f(x_i)$，则 $f(\cdot)$ 的最小二乘三角逼近式为

$$S_n(x) = \frac{1}{2}a_0 + \sum_{k=1}^{n}(a_k\cos kx + b_k\sin kx) \quad (n < m),$$

其中

$$\begin{cases} a_k = \dfrac{2}{2m+1}\displaystyle\sum_{i=0}^{2m} f_i \cos\frac{2\pi ik}{2m+1}, \\ b_k = \dfrac{2}{2m+1}\displaystyle\sum_{i=0}^{2m} f_i \sin\frac{2\pi ik}{2m+1}. \end{cases} \tag{3.9.5}$$

$(k=0,1,\cdots,n)$ 应用式(3.8.13)的结果和正交性(3.9.4)可得

$$\begin{aligned} \delta_n^2 &= \sum_{i=0}^{2m}[f_i - S_n(x_i)]^2 \\ &= \sum_{i=0}^{2m} f_i^2 - \frac{2m+1}{2}\left[\frac{a_0^2}{2} + \sum_{k=1}^{n}(a_k^2 + b_k^2)\right]. \end{aligned} \tag{3.9.6}$$

当 $n=m$ 时，最小二乘逼近就成为三角插值

$$S_m(x) = \frac{1}{2}a_0 + \sum_{k=1}^{m}(a_k\cos kx + b_k\sin kx),$$

其中系数 a_k, b_k 仍由式(3.9.5)给出，可以证明

$$S_m(x_i) = f_i \quad (i = 0, 1, \cdots, 2m),$$

从而 $\delta_m^2 = 0$，于是由式(3.9.6)得

$$\frac{2m+1}{2}\left[\frac{a_0^2}{2} + \sum_{k=1}^{m}(a_k^2 + b_k^2)\right] = \sum_{i=0}^{2m} f_i^2.$$

例 3.9.1 用下列给出的数据表求 $n=1,2,3$ 的傅里叶最小二乘逼近.

x_i	0	$2\pi/9$	$4\pi/9$	$2\pi/3$	$8\pi/9$
f_i	3.0004	5.7203	3.1993	−1.0981	−0.8679

x_i	$10\pi/9$	$4\pi/3$	$14\pi/9$	$16\pi/9$
f_i	2.9890	4.0983	1.1477	−0.1882

解 由公式(3.9.5)可求得

$$a_0 = 4.00022,\quad a_1 = 0.99998,$$
$$a_2 = 0.00011,\quad a_3 = 0.00023,$$
$$b_1 = 0.00029,\quad b_2 = 2.99997,\quad b_3 = 0.00002$$

用公式(3.9.6)可算出

$$\delta_0^2 = 44.99932,\quad \delta_1^2 = 40.49950,$$
$$\delta_2^2 = 0.00031,\quad \delta_3^2 = 0.00031.$$

通常可用$\frac{\delta_n^2}{2(m-n)}$的最小者确定 n 的值,在此例中,显然 $n=2$ 给出的最小平方逼近最好,故可得傅里叶最小二乘逼近为

$$S_2(x) = 2.00011 + 0.99998\cos x + 0.00029\sin x + 0.00011\cos 2x + 2.99997\sin 2x.$$

事实上,前面给出的数据是由

$$f(x) = 2 + \cos x + 3\sin 2x$$

的摄动得到的,而 $S_2(x)$确实能较好地逼近 $f(x)$.

更一般的情形,假定 $f(\cdot)$是以 2π 为周期的复函数,给定在 N 个点 $x_j=\frac{2\pi}{N}j\,(j=0,1,\cdots,N-1)$上的值 $f_j=f\left(\frac{2\pi}{N}j\right)$,由于

$$e^{ijx} = \cos jx + i\sin jx \quad (i=\sqrt{-1}, j = 0,1,\cdots,N-1),$$

故函数族

$$1, e^{ix}, \cdots, e^{i(N-1)x}$$

在$[0,2\pi]$上是正交的. 函数 e^{ikx} 在等距点集$\left\{x_j=\frac{2\pi}{N}j\right\}$上的值 e^{ikx_j} 组成的向量,记作

$$\boldsymbol{\phi}_k = (1, \mathrm{e}^{\mathrm{i}k\frac{2\pi}{N}}, \cdots, \mathrm{e}^{\mathrm{i}k\frac{2\pi}{N}(N-1)})^{\mathrm{T}} \quad (k = 0,1,\cdots,N-1). \tag{3.9.7}$$

N 个向量$\boldsymbol{\phi}_0,\boldsymbol{\phi}_1,\cdots,\boldsymbol{\phi}_{N-1}$也是正交的，即

$$\begin{aligned}(\boldsymbol{\phi}_k,\boldsymbol{\phi}_l) &= \sum_{j=0}^{N-1} \mathrm{e}^{\mathrm{i}k\frac{2\pi}{N}j}\mathrm{e}^{-\mathrm{i}l\frac{2\pi}{N}j} \\ &= \sum_{j=0}^{N-1} \mathrm{e}^{\mathrm{i}(k-l)\frac{2\pi}{N}j} \\ &= \begin{cases}0, l \neq k, \\ N, l = k.\end{cases}\end{aligned} \tag{3.9.8}$$

因此，$f(x)$在 N 个点$\{x_j\}$上的有限傅里叶展开式为

$$S(x) = \sum_{k=0}^{n-1} c_k \mathrm{e}^{\mathrm{i}kx} \quad (n \leqslant N), \tag{3.9.9}$$

其中

$$c_k = \frac{1}{N}\sum_{j=0}^{N-1} f_j \mathrm{e}^{-\mathrm{i}kj\frac{2\pi}{N}} \quad (k = 0,1,\cdots,N-1). \tag{3.9.10}$$

在式(3.9.9)中，$n=N$ 时 $S(x)$即为 $f(x)$在点$\{x_j\}$上的插值函数. 此时有

$$f_j = \sum_{k=0}^{N-1} c_k \mathrm{e}^{\mathrm{i}k\frac{2\pi}{N}j} \quad (j = 0,1,\cdots,N-1). \tag{3.9.11}$$

式(3.9.10)称为**离散傅里叶变换**(discrete fourier transform, DFT)；而式(3.9.11)则称为**反变换**. DFT(3.9.10)也是傅里叶变换

$$H(x) = \int_{-\infty}^{+\infty} f(t)\mathrm{e}^{-2\pi\mathrm{i}xt}\,\mathrm{d}t$$

离散化的结果，DFT 是用计算机进行傅里叶分析的主要方法，它在数字讯号处理、全息技术、光谱和声谱分析等很多领域都有广泛应用.

3.9.2 快速傅里叶变换

无论用公式(3.9.5)计算傅里叶逼近的系数 a_k 及 b_k，还是用公式(3.9.10)及公式(3.9.11)计算 DFT，都可归结为计算

$$c_k = \sum_{j=0}^{N-1} x_j \omega^{kj} \quad (k = 0,1,\cdots,N-1), \qquad (3.9.12)$$

其中 $\omega = \mathrm{e}^{-\mathrm{i}\frac{2\pi}{N}}$ 或 $\omega = \mathrm{e}^{\mathrm{i}\frac{2\pi}{N}}$，$\{x_j\}(j=0,1,\cdots,N-1)$ 是已知复数序列．如直接用式(3.9.12)计算一个 c_k 要用 N 次复数乘法和 N 次复数加法，称为 N 个操作，计算全部 c_k 共要 N^2 个操作，当 N 很大时运算量也很大，大量数据处理时使用高速计算机有时也无法计算，直到 60 年代中期产生了适合计算机使用的快速傅里叶变换(FFT)算法，才大大提高了运算速度．FFT 算法的基本思想是尽量减少乘法的运算次数．例如，$ab+ac+ad=a(b+c+d)$，左端三次乘法而右端只用一次乘法，实际上

$$\mathrm{e}^{\mathrm{i}\frac{2\pi}{N}kj} = \cos\frac{2\pi}{N}kj + \mathrm{i}\sin\frac{2\pi}{N}kj \quad (k,j = 0,1,\cdots,N-1)$$

中只有 N 个不同的值，特别当 $N=2^p$ 时，只有 $\frac{N}{2}$ 个不同的值，因此，在计算式(3.9.12)时，可合并大量同因子的项，从而减少乘法次数．以 $N=2^3=8$ 为例．由式(3.9.12)，当 $N=8$ 时，得

$$c_k = \sum_{j=0}^{7} x_j \omega^{kj} \quad (k = 0,1,\cdots,7). \qquad (3.9.13)$$

将 j,k 用二进制表示为

$$j = j_2 2^2 + j_1 2^1 + j_0 2^0 = (j_2 j_1 j_0),$$
$$k = k_2 2^2 + k_1 2^1 + k_0 2^0 = (k_2 k_1 k_0),$$

其中 $j_r, k_r (r=0,1,2)$ 只能取 0 或 1，此时 $0 \leqslant j,k \leqslant 7$．相应地，记

$$c_k = C(k_2 k_1 k_0), \quad x_j = X(j_2 j_1 j_0),$$

于是公式(3.9.13)可表示为

$$C(k_2 k_1 k_0) = \sum_{j_0=0}^{1}\sum_{j_1=0}^{1}\sum_{j_2=0}^{1} x(j_2 j_1 j_0)\omega^{(k_2 k_1 k_0)(j_2 j_1 j_0)}$$
$$= \sum_{j_0=0}^{1}\left\{\sum_{j_1=0}^{1}\left[\sum_{j_2=0}^{1} x(j_2 j_1 j_0)\omega^{k_0(j_2 j_1 j_0)}\right]\omega^{k_1(j_1 j_0 0)}\right\}\omega^{k_2(j_0 00)}.$$

若引入记号

$$A_0(j_2j_1j_0)=x(j_2j_1j_0),$$

$$\begin{cases}A_1(j_1j_0k_0)=\sum\limits_{j_2=0}^{1}A_0(j_2j_1j_0)\omega^{k_0(j_2j_1j_0)},\\ A_2(j_0k_1k_0)=\sum\limits_{j_1=0}^{1}A_1(j_1j_0k_0)\omega^{k_1(j_1j_00)},\\ A_3(k_2k_1k_0)=\sum\limits_{j_0=0}^{1}A_2(j_0k_1k_0)\omega^{k_2(j_000)},\end{cases}\tag{3.9.14}$$

则

$$C(k_2k_1k_0)=A_3(k_2k_1k_0).$$

说明计算 $c_k=C(k_2k_1k_0)$ 可分为 3 步进行；对一般 $N=2^p$ 的情形，则计算 c_k 可分为 p 步进行，注意到

$$\omega^{k_02^{p-1}}=\omega^{k_0\frac{N}{2}}=(-1)^{k_0},$$

公式(3.9.14)还可化简为

$$\begin{aligned}A_1(j_1j_0k_0)&=\sum_{j_2=0}^{1}A_0(j_2j_1j_0)\omega^{k_0(j_2j_1j_0)}\\&=A_0(0j_1j_0)\omega^{k_0(0j_1j_0)}+A_0(1j_1j_0)\omega^{k_02^2}\omega^{k_0}(0j_1j_0)\\&=[A_0(0j_1j_0)+(-1)^{k_0}A_0(1j_1j_0)]\omega^{k_0(0j_1j_0)},\end{aligned}$$

即

$$\begin{cases}A_1(j_1j_00)=A_0(0j_1j_0)+A_0(1j_1j_0),\\ A_1(j_1j_01)=[A_0(0j_1j_0)-A_0(1j_1j_0)]\omega^{(0j_1j_0)}.\end{cases}$$

将二进制还原为十进制表示 $j=(0j_1j_0)=j_12^1+j_02^0$，即 $j=0,1,2,3$ 时得

$$\begin{cases}A_1(2j)=A_0(j)+A_0(j+2^2),\\ A_1(2j+1)=[A_0(j)-A_0(j+2^2)]\omega^j,\end{cases}\quad j=0,1,2,3.\tag{3.9.15}$$

同理，式(3.9.14)中的 A_2 可简化为

$$
\begin{cases}
A_2(j2^2 + k) = A_1(2j + k) + A_1(2j + k + 2^2), \\
A_2(j2^2 + k + 2) = [A_1(2j + k) - A_1(2j + k + 2^2)]\omega^{2j}, \\
\qquad\qquad k, j = 0, 1,
\end{cases}
\tag{3.9.16}
$$

式(3.9.14)中的 A_3 可简化为

$$
\begin{cases}
A_3(k) = A_2(k) + A_2(k + 2^2), \\
A_3(k + 2^2) = A_2(k) - A_2(k + 2^2),
\end{cases}
\quad k = 0, 1, 2, 3.
\tag{3.9.17}
$$

根据公式(3.9.15)～(3.9.17)，由 $A_0(j)=x(j)(j=0,1,\cdots,7)$ 出发逐次计算到 $A_3(k)=c_k$，就得到所要的结果，总共只做 8 次复数乘法就求出全部 $c_k(k=0,\cdots,7)$.

将上面得到的 $N=2^3$ 的计算公式(3.9.15)～(3.9.17)推广到 $N=2^p$ 的一般情形可得 FFT 的计算公式如下：

$$
\begin{cases}
A_q(j2^q + k) = A_{q-1}(j2^{q-1} + k) + A_{q-1}(j2^{q-1} + k + 2^{p-1}), \\
A_q(j2^q + k + 2^{q-1}) = [A_{q-1}(j2^{q-1} + k) - A_{q-1}(j2^{q-1} + \\
\qquad\qquad k + 2^{p-1})]\omega^{j2^{q-1}}.
\end{cases}
\tag{3.9.18}
$$

其中 $q=1,\cdots,p$；$j=0,1,\cdots,2^{p-q}-1$；$k=0,1,\cdots,2^{q-1}-1$.
这里 A_q 括号内的数代表序号，亦即在计算机中存放该数的地址号；一组 A_q 占用 N 个复数单元，计算时只需用两组单元轮流替换，由 $A_0(j)=x(j)(j=0,1,\cdots,N-1)$ 出发，$q=1,\cdots,p$，用公式(3.9.18)计算到 $A_p(k)=c_k \quad (k=0,1,\cdots,N-1)$ 即为所求. 在计算机上的计算步骤见算法 3.9.2，计算一组 A_q 共做 $2^{p-q}\cdot 2^{q-1}=2^{p-1}=\dfrac{N}{2}$ 次复数乘法，而最后一步计算 A_p 时，由于 $\omega^{j2^{p-1}}=(\omega^{\frac{N}{2}})^j=(-1)^j=(-1)^0=1$（注意：当 $q=p$ 时 $2^{p-q}-1=0$，故 $j=0$），因此，当 $q=p$ 时计算 A_p 不用乘法，故共需要计算 $(p-1)\dfrac{N}{2}$ 次复数乘法，这比直接用公式(3.9.12)计算需 N^2 次复数乘法少得多，其

比值为$\frac{p-1}{2}$：N. 如当 $N=2^{10}$ 时为 4.5：1024≈1：230；它比通常的 FFT 算法用 Np 次复数乘法也快一倍以上. 因此，计算公式(3.9.18)称为改进的 FFT 算法，其计算程序步骤如下：

算法 3.9.2 改进的 FFT

1. 给出复数数组 $A_0(N)$，$A_1(N)$ 及 $\omega\left(\frac{N}{2}\right)$，确定 p. 将已知数据$\{x_j\}$输入到数组 $A_0(N)$ 的单元中.

2. 计算 $\omega^m=e^{-i\frac{2\pi}{N}m}$(或 $\omega^m=e^{i\frac{2\pi}{N}m}$)($m$ 从 0 到 $2^{p-1}-1$)，结果存放在 $\omega(m)$ 的相应单元中.

3. j 从 0 到$(2^{p-q}-1)$，k 从 0 到$(2^{q-1}-1)$按公式(3.9.18)计算 A_q，结果存放在 A_1 的相应单元. 即求

$$A_1(j2^q+k)=A_0(j2^{q-1}+k)+A_0(j2^{q-1}+k+2^{p-1}),$$

$$A_1(j2^q+k+2^{q-1})=[A_0(j2^{q-1}+k)-A_0(j2^{q-1}+k+2^{p-1})]\omega(j2^{q-1}),$$

圆括号内的数表示地址号，即数据存放位置.

4. k，j 循环结束，$q+1\to q$，$A_1(N)\to A_0(N)$. 若 $q<p$，则转步骤 3，否则执行步骤 5.

5. 此时 $q=p$，$j=0$，k 从 0 到$(2^{p-1}-1)$，计算

$$A_1(k)=A_0(k)+A_0(k+2^{p-1}),$$

$$A_1(k+2^{p-1})=A_0(k)-A_0(k+2^{p-1}),$$

这里得到的 $A_1(k)$($k=0,1,\cdots,2^p-1$)即为所求 c_k.

3.10 有理逼近与连分式

多项式是函数逼近的一种很好的工具，但用多项式逼近计算函数值并不是最好的，特别当函数具有极点时，用有理函数逼近比用多项式逼近要精确得多. 所谓有理函数是指形如

$$R_{mn}(x)=\frac{P_m(x)}{Q_n(x)}=\frac{a_m x^m+\cdots+a_1 x+a_0}{b_n x^n+\cdots+b_1 x+b_0} \tag{3.10.1}$$

的有理分式，其中 P_m 与 Q_n 分别为 m 次和 n 次多项式. 虽然有理函数计算要用除法运算，但在计算机上除法与乘法的运算时间大体相同，而有理逼近又可减少乘除法运算次数，因此，计算机上使用的函数子程序有很多都用有理逼近. 用式(3.10.1)的有理逼近其计算量粗略地等同于 $m+n$ 次的多项式逼近，但其误差通常比多项式逼近小. 连分式在有理逼近与有理分式计算中起重要作用. 如在计算 $R_{mn}(x)$中，化为连分式时只要用 m 或 n 次乘除法运算，比 $m+n$ 次多项式用 $m+n$ 次乘法运算减少了运算量.

展开式

$$b_0+\cfrac{a_1}{b_1+\cfrac{a_2}{b_2+\ddots+\cfrac{a_n}{b_n+\ddots}}} \tag{3.10.2}$$

称为**连分式**(continued fraction)，它可表示为

$$b_0+\frac{a_1}{b_1}+\frac{a_2}{b_2}+\cdots+\frac{a_n}{b_n}+\cdots$$

分式$\frac{a_n}{b_n}$称为连分式(3.10.2)的第 n 节，a_n，b_n 称为连分式(3.10.2)第 n 节的两项，$a_1,\cdots,a_n,\cdots$叫做连分式的部分分子，$b_1,\cdots,b_n,\cdots$叫做连分式的部分分母. 假定所有 b_i 均不为零，有限连分式

$$b_0+\frac{a_1}{b_1}+\cdots+\frac{a_n}{b_n}=\frac{P_n}{Q_n}$$

称为连分式(3.10.2)的第 n 个渐近分式. 相邻三个渐近分式之间有递推关系：

$$\begin{cases}P_n=b_nP_{n-1}+a_nP_{n-2},\\ Q_n=b_nQ_{n-1}+a_nQ_{n-2}.\end{cases} \tag{3.10.3}$$

事实上，按连分式定义有

$$\frac{P_0}{Q_0}=\frac{b_0}{1},\quad \frac{P_1}{Q_1}=b_0+\frac{a_1}{b_1}=\frac{b_0b_1+a_1}{b_1},$$

$$\frac{P_2}{Q_2}=b_0+\frac{a_1}{b_1+}\frac{a_2}{b_2}=b_0+\frac{a_1b_2}{b_1b_2+a_2}=\frac{b_2P_1+a_2P_0}{b_2Q_1+a_2Q_0},$$

若取 $P_{-1}=1, Q_{-1}=0$，则当 $n=1$ 和 $n=2$ 时式(3.10.3)成立.假设式(3.10.3)对 n 成立，要证明它对 $n+1$ 也成立，只要注意由$\frac{P_n}{Q_n}$推导$\frac{P_{n+1}}{Q_{n+1}}$时，用 $b_n+\frac{a_{n+1}}{b_{n+1}}$代替 b_n 即可得到

$$\begin{aligned}\frac{P_{n+1}}{Q_{n+1}}&=\frac{b_nP_{n-1}+\frac{a_{n+1}}{b_{n+1}}P_{n-1}+a_nP_{n-2}}{b_nQ_{n-1}+\frac{a_{n+1}}{b_{n+1}}Q_{n-1}+a_nQ_{n-2}}\\&=\frac{b_{n+1}(b_nP_{n-1}+a_nP_{n-2})+a_{n+1}P_{n-1}}{b_{n+1}(b_nQ_{n-1}+a_nQ_{n-2})+a_{n+1}Q_{n-1}}\\&=\frac{b_{n+1}P_n+a_{n+1}P_{n-1}}{b_{n+1}Q_n+a_{n+1}Q_{n-1}},\end{aligned}$$

从而证明了式(3.10.3)对一切 n 成立.

例 3.10.1 $\ln(1+x)$ 当 $x>-1$ 时有如下的展开式：

$$\ln(1+x)=\frac{x}{1+}\frac{x}{2+}\frac{x}{3+}\frac{2^2x}{4+}\frac{2^2x}{5+}\frac{3^2x}{6+}\frac{3^2x}{7+}\cdots$$

将它化为有理分式.

解 先计算

$$\frac{P_0}{Q_0}=\frac{x}{1},\quad \frac{P_1}{Q_1}=\frac{x}{1+}\frac{x}{2}=\frac{2x}{2+x},$$

再由公式(3.10.3)求得

$$\frac{P_2}{Q_2}=\frac{3(2x)+x(x)}{3(x+2)+x\cdot 1}=\frac{6x+x^2}{6+4x},$$

$$\frac{P_3}{Q_3}=\frac{4(6x+x^2)+4x(2x)}{4(6+4x)+4x(2+x)}=\frac{4(6x+3x^2)}{4(6+6x+x^2)},$$

$$\frac{P_4}{Q_4}=\frac{20(6x+3x^2)+4x(6x+x^2)}{20(6+6x+x^2)+4x(6+4x)}=\frac{4(30x+21x^2+x^3)}{4(30+36x+9x^2)},$$

$$\frac{P_5}{Q_5}=\frac{4[6(30x+21x^2+x^3)+9x(6x+3x^2)]}{4[6(30+36x+9x^2)+9x(6+6x+x^2)]}$$

$$=\frac{12(60x+60x^2+11x^3)}{12(60+90x+36x^2+3x^3)},$$

注意：利用递推公式(3.10.3)计算时分子与分母的公因数是不能随便约去的，但作为最后结果的有理分式可以约去公因数.

利用 $\ln(1+x)$ 的连分式展开，可获得前 4 个有理逼近：

$$\begin{cases} R_{11}(x)=\dfrac{2x}{2+x}, \\ R_{22}(x)=\dfrac{6x+3x^2}{6+6x+x^2}, \\ R_{33}(x)=\dfrac{60x+60x^2+11x^3}{60+90x+36x^2+3x^3}, \\ R_{44}(x)=\dfrac{420x+630x^2+260x^3+25x^4}{420+840x+540x^2+120x^3+6x^4}. \end{cases}$$

$\ln(1+x)$ 的泰勒展开式前 $2n$ 项和

$$p_{2n}(x)=x-\frac{x^2}{2}+\frac{x^3}{3}-\cdots+(-1)^{2n-1}\frac{x^{2n}}{2n}\quad(-1<x\leqslant 1).$$

$R_{nn}(x)$ 与 $p_{2n}(x)$ 有相同个数的参数，但它们逼近 $\ln(1+x)$ 的精度差别很大，在用 $R_{nn}(1)$ 与 $p_{2n}(1)$ 作为 $\ln 2$ 的近似时，其误差 ϵ_R 与 ϵ_p 的大小情况如下表：

n	$R_{nn}(1)$	ϵ_R	$p_{2n}(1)$	ϵ_p
1	0.667	0.026	0.50	0.19
2	0.69231	0.00084	0.58	0.11
3	0.693122	0.000025	0.617	0.076
4	0.69314642	0.00000076	0.634	0.058

$\ln 2$ 的精确值为 $\ln 2=0.69314718\cdots$，$R_{44}(1)$ 与 $p_8(1)$ 的精度相差

十万倍，这说明某些函数用有理逼近是很必要的.

将有理分式(3.10.1)化为连分式，使乘除运算次数减少. 对 $m=n$ 的情形，可得

$$R_{nn}(x)=\frac{a_nx^n+\cdots+a_1x+a_0}{b_nx^n+\cdots+b_1x+b_0}$$

$$=\frac{a_n}{b_n}+\cfrac{c_1}{x+\cfrac{a_{n-1}^{(1)}x^{n-1}+\cdots}{b_{n-1}^{(1)}x^{n-1}+\cdots}}$$

$$=c_0+\frac{c_1}{x+R_{n-1,n-1}(x)},$$

这里 c_0, c_1 及 $R_{n-1,n-1}(x)$ 可通过除法得到，对 $R_{n-1,n-1}(x)$ 再按同样步骤进行，最后可得

$$R_{nn}(x)=c_0+\frac{c_1}{x+B_1}+\frac{c_2}{x+B_2}+\cdots+\frac{c_n}{x+B_n}. \tag{3.10.4}$$

用公式(3.10.4)计算 $R_{nn}(x)$ 的值只用 n 次除法运算，若直接用有理分式计算则要用 $2n$ 次乘除法运算.

例 3.10.2 将 $R_{33}(x)=\dfrac{x^3+3x^2-7x-28}{x^3-2x^2-2x-3}$ 化为连分式.

解

$$R_{33}(x)=1+\frac{5(x^2-x-5)}{x^3-2x^2-2x-3}$$

$$=1+\cfrac{5}{x-1+\cfrac{2x-8}{x^2-x-5}}$$

$$=1+\cfrac{5}{x-1+\cfrac{2}{\cfrac{x^2-x-5}{x-4}}}$$

$$=1+\frac{5}{x-1}+\frac{2}{x+3}+\frac{7}{x-4}.$$

如果有理分式(3.11.1)中 $m>n$,令 $m=n+k$,则可将它表成

$$R_{mn}(x) = d_0 + \cdots + d_k x^k + \frac{c_1}{x+B_1 + \cdots + \dfrac{c_n}{x+B_n}}. \tag{3.10.5}$$

用公式(3.10.5)计算 $R_{mn}(x)$的值需用 $n+k=m$ 次乘除法运算.

若 $m<n$,令 $n=m+k$,则

$$R_{mn}(x) = \frac{c_0}{d_0 + \cdots + d_{k-1}x^{k-1} + x^k + \dfrac{c_1}{x+B_1 + \cdots + \dfrac{c_m}{x+B_m}}}. \tag{3.10.6}$$

用公式(3.10.6)计算 $R_{mn}(x)$的值也需 $k+m=n$ 次乘除法运算.因此,无论 m,n 为何种情形,将 $R_{mn}(x)$化为连分式计算只需$\max(n,m)$次乘除法运算,比直接计算 $R_{mn}(x)$时用 $n+m$ 次乘除法运算节省计算量,比相应的 $m+n$ 次多项式逼近的乘除法运算次数也少.

3.11 最佳有理逼近

给定函数 $f(\cdot)\in C[a,b]$及一对整数 $m\geqslant 0$ 及 $n\geqslant 0$,用有理分式

$$R_{mn}(x) = \frac{P_m(x)}{Q_n(x)} \tag{3.11.1}$$

逼近 $f(\cdot)$,这里 $P_m(\cdot)\in H_m$,$Q_n(\cdot)\in H_n$ 是次数不超过 m 和 n 的多项式.设 $P_m(\cdot)$与 $Q(\cdot)$互质,用 $R_n^m[a,b]$表示有理函数类

$$R_n^m[a,b] = \left\{\frac{P_m}{Q_n},\quad P_m \in H_m,\quad Q_n \in H_n \text{ 在}[a,b]\text{上 } Q_n > 0\right\}.$$

定义 3.11.1 量

$$\Delta(R_{mn}) = \sup_{a\leqslant x\leqslant b} | f(x) - R_{mn}(x) |$$

称为 $f(\cdot)$与 $R_{mn}(\cdot)$的偏差.在 $R_n^m[a,b]$中的最小偏差记作

$$\begin{aligned}\rho_{mn}(f) &= \inf_{R_n^m[a,b]} \Delta(R_{mn}) \\ &= \inf_{R_n^m} \sup_{x\in[a,b]} | f(x) - R_{mn}(x) |,\end{aligned} \tag{3.11.2}$$

称为函数 $f(\cdot)$在有理函数类 $R_n^m[a,b]$中的**最佳有理逼近**(best rational approximations). 若存在 $R(x)\in R_n^m[a,b]$,使得

$$\Delta(R)=\rho_{mn}(f),$$

则称 $R(x)$为 $f(\cdot)$在有理函数类 $R_n^m[a,b]$中的**最佳逼近有理分式**.

定理 3.11.2 设 $f(\cdot)\in C[a,b]$,则在有理函数类 $R_n^m[a,b]$中至少存在一个最佳逼近有理分式 $R(x)$.

定理表明连续函数 $f(\cdot)$在 $R_n^m[a,b]$中最佳逼近有理分式的存在性,通常可将 $R(x)$写成

$$R(x)=\frac{a_0+a_1x+\cdots+a_{m-\mu}x^{m-\mu}}{b_0+b_1x+\cdots+b_{n-\nu}x^{n-\nu}}=\frac{A(x)}{B(x)},\qquad(3.11.3)$$

其中 $A(\cdot)$与 $B(\cdot)$互质,$b_{n-\nu}\neq 0$,$a_{m-\mu}\neq 0$,$0\leqslant\mu\leqslant m$,$0\leqslant\nu\leqslant n$. 记 $d=\min(\mu,\nu)$,称为 $R(\cdot)$的**亏项数**. 如果 $d>0$,则称 $R(\cdot)$是**退化的**. 与多项式的最佳一致逼近切比雪夫定理类似,对最佳逼近有理分式也有相应定理.

定理 3.11.3 设 $f\in C[a,b]$,在形如式(3.11.1)的有理函数类 $R_n^m[a,b]$中,有理分式(3.11.3)是最佳逼近有理分式的充要条件是:在$[a,b]$上至少有 $N=m+n-d+2$ 个点,使 $f(x)-R(x)$以正负交错的符号达到 $\Delta(R)$. 其中 d 是 $R(\cdot)$的亏项数,满足上述条件的最佳逼近有理分式 $R(x)$是惟一确定的(惟一是指化简后为相同有理分式).

根据这定理,对 $\mu=\nu=0$ 的非退化情形,求形如

$$R(x)=\frac{a_0+a_1x+\cdots+a_mx^m}{1+b_1x+\cdots+b_nx^n}\quad(b_0=1)\qquad(3.11.4)$$

的最佳逼近有理分式,可从一个初始近似 $R^{(0)}(x)$出发,用逐次逼近的有理分式序列 $R^{(r)}(x)$求得 $R(x)$的足够精确的近似. 具体实现可用列梅兹算法:

(1) 给定初始近似 $R^{(0)}(x)$,在$[a,b]$中选取 $N=m+n+2$ 个点组成的点集 $X_1=\{x_i^{(1)},i=1,\cdots,N\}$,排列为

$$a\leqslant x_1^{(1)}\leqslant x_2^{(1)}\leqslant\cdots\leqslant x_N^{(1)}\leqslant b.$$

它使 $f(x_i^{(1)})-R^{(0)}(x_i^{(1)})$ 正负交错，这 N 个点一般就是 $f(x)-R^{(0)}(x)$ 的极值点.

(2) 假设已求出点集 $X_r=\{x_i^{(r)}, i=1,\cdots,N\}$，求 r 次近似

$$R^{(r)}(x)=\frac{a_0^{(r)}+a_1^{(r)}x+\cdots+a_m^{(r)}x^m}{1+b_1^{(r)}x+\cdots+b_n^{(r)}x^n},$$

它的系数可通过求解非线性方程组

$$f(x_i^{(r)})-R^{(r)}(x_i^{(r)})=(-1)^iE \quad (i=1,2,\cdots,N) \tag{3.11.5}$$

得到.

(3) 求取得 $\max\limits_{a\leqslant x\leqslant b}|f(x)-R^{(r)}(x)|$ 的点 $\tau\in[a,b]$.

(4) 用 τ 取代点集 X_r 中的某一点，使得 $f(x)-R^{(r)}(x)$ 在新求得的点集 $X_{r+1}=\{x_i^{(r+1)}, i=1,\cdots,N\}$ 上交错变号. 由于 $f(x)-R^{(r)}(x)$ 在点集 X_r 上交错变号，而 X_{r+1} 只改换 X_r 的一点，当 τ 落在 $x_j^{(r)}$ 与 $x_{j+1}^{(r)}$ 之间时，τ 必同 $x_j^{(r)}$ 或 $x_{j+1}^{(r)}$ 的某一点使 $f(x)-R^{(r)}(x)$ 取同号，此时以 τ 代替这个"同号"的点即可，当 τ 在端点时，可根据 $x_1^{(r)}$ 与 $x_N^{(r)}$ 上 $f(x)-R^{(r)}(x)$ 符号情况决定 τ 究竟应取代那一点. 总之，使新点集 X_{r+1} 上 $f(x)-R^{(r)}(x)$ 交错变号是可能的.

(5) 以 X_{r+1} 代替 X_r，重复第 2,3,4 步，直到 $R^{(r)}(x)$ 满足精度要求为止.

在假定初始近似 $R^{(0)}(x)$ 足够好的前提下，赖尔斯顿 (Ralston) 给出了列梅兹算法的收敛性定理.

定理 3.11.4 设 $R^*(x)$ 为 $f(x)$ 的形如式(3.11.4)的最佳一致逼近有理分式，上述列梅兹算法的初始近似 $R^{(0)}(x)$ 于 $[a,b]$ 上恰有 N 个正负相间的极值点，它的第一个极值与 $f(x)-R^*(x)$ 的第一个极值符号相同，设 $X_1=\{x_1^{(1)},\cdots,x_N^{(1)}\}$ 是 $f(x)-R^{(0)}(x)$ 的 N 个极值点的横坐标

$$a\leqslant x_1^{(1)}<x_2^{(1)}<\cdots<x_N^{(1)}\leqslant b,$$

$E_{\min}$是误差极值的最小值. 又设 $x^* = \{x_i^*\}(i=1,\cdots,N)$是 $f(x)-R^*(x)$的 N 个极值点(极值大小为 $r_{mn}^* = \| f(x)-R^*(x) \|_\infty$)的横坐标,

$$a \leqslant x_1^* < x_2^* < \cdots < x_N^* \leqslant b,$$

则有 $\varepsilon>0$ 和 $\eta>0$ 存在,使当

$$|x_i^{(1)} - x_i^*| < \varepsilon \quad (i=1,\cdots,N),$$

$$r_{mn}^* - E_{\min} < \eta$$

时,只要采用恰当的方法求解方程组(3.11.5),则列梅兹算法收敛,即 $\lim\limits_{r\to\infty} R^{(r)}(x) = R^*(x)$.

在定理中要求 $R^{(0)}(x)$具有 N 个正负相间的极值,而在实践中并无必要,应用时只要 $R^{(0)}(x)$有 N 个极值就够了. 实际上列梅兹算法只要求一个点集 X_1,通常可利用$[a,b]$上的切比雪夫多项式 $T_{m+n+1}(x)$的 $N=m+n+2$ 个极值点作为 X_1. 另一种办法是用某种有理逼近(见 3.12 节或 3.13 节)的极值点,当然这个有理逼近要有 $m+n+2$ 个极值点.

求解方程组(3.11.5)通常可用割线法,实践表明是成功的. 费凯(Fike)给出另一种迭代法,先将(3.11.5)化为等价形式

$$\begin{aligned} & a_0 + a_1 x_i^{(1)} + \cdots + a_m (x_i^{(1)})^m + \\ & b_1 x_i^{(1)} [(-1)^i E - f(x_i^{(1)})] + \cdots + \\ & b_n (x_i^{(1)})^n [(-1)^i E - f(x_i^{(1)})] + (-1)^i E \\ = & f(x_i^{(1)}) \quad (i=1,\cdots,N), \end{aligned} \tag{3.11.6}$$

再把此式改写为

$$\begin{aligned} & a_0 + a_1 x_i^{(1)} + \cdots + a_m (x_i^{(1)})^m + \\ & b_1 x_i^{(1)} [(-1)^i \bar{E} - f(x_i^{(1)})] + \cdots + \\ & b_n (x_i^{(1)})^n [(-1)^i \bar{E} - f(x_i^{(1)})] + (-1)^i E \\ = & f(x_i^{(1)}) \quad (i=1,\cdots,N), \end{aligned} \tag{3.11.7}$$

取 $\bar{E}=0$,则式(3.11.7)是关于 $a_0,\cdots,a_m,b_1,\cdots,b_n$ 和 E 的线性方程组,可求出 $a_0,\cdots,a_m,b_1,\cdots,b_n$ 和 $E=E^{(1)}$ 的解.以新的 $E^{(1)}$ 代替方程(3.11.7)的 $\bar{E}$,则又可求出 $a_0,\cdots,a_m,b_1,\cdots,b_n$ 和 $E=E^{(2)}$;又以 $E^{(2)}$ 代替 $\bar{E}$,再解出一组 $a_0,\cdots,a_m,b_1,\cdots,b_n$ 和 $E^{(3)}$;依此类推.费凯证明了该过程收敛.

例 3.11.5 设 $f(x)=\cos\frac{1}{4}\pi x$,在$[-1,1]$上用

$$R(x)=\frac{a_0+a_1x+a_2x^2+a_3x^3+a_4x^4+a_5x^5}{1+b_1x+b_2x^2}$$

作最佳一致逼近.

解 因 $f(x)$是偶函数,$[-1,1]$是关于原点对称的区间,所以 $R^*(x)$的分子与分母必只出现偶次幂,故只需考虑有理分式

$$R(x)=\frac{a_0+a_2x^2+a_4x^4}{1+b_2x^2}.$$

由切比雪夫逼近理论,满足定理 3.11.3 的交错点集应有对称的 9 个点,它们是

$$-x_1^*=x_9^*,\quad -x_2^*=x_8,\quad -x_3^*=x_7^*,$$
$$-x_4^*=x_6^*,\quad x_5^*=0,$$

故在列梅兹算法第 1 步中,点集 X_1 仍应保持这一关系,可取

$$x_i^{(1)}=\cos\frac{(9-i)}{8}\pi\quad(i=5,6,7,8,9),$$

相应式(3.11.6)的非线性方程组为

$$\begin{aligned}&a_0+a_2(x_i^{(1)})^2+a_4(x_i^{(1)})^4+\\&b_2(x_i^{(1)})^2\left((-1)^iE-\cos\frac{\pi}{4}x_i^{(1)}\right)-E\\=&f(x_i^{(1)})\quad(i=5,6,7,8,9).\end{aligned}$$

用 Remez 算法只需迭代两次,即可求得

$$a_0^*=1.0000000241,\quad a_2^*=-0.2874648358,$$
$$a_4^*=0.0093933390,\quad b_2^*=0.0209610796.$$

其偏差为 $E=0.241\times10^{-7}$,

$$\cos\frac{\pi}{4}x$$

$$\approx\frac{1.0000000241-0.2874648358x^2+0.009393339x^4}{1+0.0209610796x^2}.$$

3.12 帕德逼近

帕德逼近是以函数的幂级数展开为基础，设 $f(\cdot)$ 在原点邻域内可展成幂级数

$$f(x)=\sum_{j=0}^{\infty}a_jx^j, \tag{3.12.1}$$

再设 $f(\cdot)$ 的有理逼近为

$$R_{LM}(x)=\frac{P_L(x)}{Q_M(x)}=\frac{\sum_{k=0}^{L}p_kx^k}{1+\sum_{k=1}^{M}q_kx^k}, \tag{3.12.2}$$

其中 $P_L(\cdot)\in H_L, Q_M(\cdot)\in H_M, p_k$ 及 q_k 均为待定系数，可由下列方程确定

$$f(x)-\frac{P_L(x)}{Q_M(x)}=O(x^{L+M+1}). \tag{3.12.3}$$

用 $Q_M(x)$ 乘(3.12.3)式，并将 $f(x)$ 用(3.12.1)表示，然后比较两端 x^j 同次幂系数，即得关于系数 $p_0,\cdots,p_L$ 及 $q_1,\cdots,q_M$ 的线性方程组

$$\begin{cases}a_0 & =p_0,\\ a_1+a_0q_1 & =p_1,\\ a_2+a_1q_1+a_0q_2 & =p_2,\\ \vdots & \\ a_L+a_{L-1}q_1+\cdots+a_{L-M}q_M & =p_L,\\ a_{L+1}+a_Lq_1+\cdots+a_{L-M+1}q_M & =0,\\ \vdots & \\ a_{L+M}+a_{L+M-1}q_1+\cdots+a_Lq_M & =0,\end{cases} \tag{3.12.4}$$

其中已规定 $a_n=0$. 当 $n<0$ 时,这是关于 $p_0,\cdots,p_L,q_1,\cdots,q_M$ 的 $L+M+1$个未知量的线性方程组,方程个数是 $L+M+1$,求解时可由最后的 M 个方程求出 $q_1,\cdots,q_M$,把值代入前 $L+1$ 个方程,即可求得 $p_0,p_1,\cdots,p_L$. 因此,对任何给定的非负整数 L,M,都可由方程组(3.12.4)求出 $f(x)$的有理逼近 $R_{LM}(x)=\dfrac{P_L(x)}{Q_M(x)}$,这里 $Q_M(0)=1$. 这样的有理分式函数 R_{LM} 即为 f 的(L,M)阶 Padé 逼近,用$[L/M]$表示.

定理 3.12.1 对任意形式的幂级数 $f(x)$,若其帕德逼近存在,则必惟一.

根据惟一性定理,当方程组(3.12.4)的解存在时,不论用什么方法求得满足(3.12.4)的解 $P_L(x)$与 $Q_M(x)$,由它产生的有理分式 $R_{LM}(x)$都是所要求的帕德逼近$[L/M]$,通常可用被称作"Padé 表"的图表(见表 3.2)来表示$[L/M]$的位置.

表 3.2 Padé 表

L \ M	0	1	2	3
0	[0/0]	[0/1]	[0/2]	[0/3]…
1	[1/0]	[1/1]	[1/2]	[1/3]…
2	[2/0]	[2/1]	[2/2]	[2/3]…
3	[3/0]	[3/1]	[3/2]	[3/3]…
⋮	⋮	⋮	⋮	⋮

当 $M=0$ 时,就是表中第一列为 f 的泰勒展开. 若

$$f(x)=\mathrm{e}^x=1+x+\frac{x^2}{2!}+\frac{x^3}{3!}+\cdots$$

则有 e^x 的 Padé 表(见表 3.3).

表 3.3　e^x 的 Padé 表

L \ M	0	1	2	3	4
0	$\frac{1}{1}$	$\frac{1}{1-x}$	$\frac{2}{2-2x+x^2}$	$\frac{6}{6-6x+3x^2-x^3}$	$\frac{24}{24-24x+12x^2-4x^3+x^4}$
1	$\frac{1+x}{1}$	$\frac{2+x}{2-x}$	$\frac{6+2x}{6-4x+x^2}$	$\frac{24+6x}{24-18x+6x^2-x^3}$	$\frac{120+24x}{120-90x+36x^2-8x^3+x^4}$
2	$\frac{2+2x+x^2}{2}$	$\frac{6+4x+x^2}{6-2x}$	$\frac{12+6x+x^2}{12-6x+x^2}$	$\frac{60+24x+3x^2}{60-36x+9x^2-x^3}$	$\frac{360+120x+12x^2}{360-240x+72x^2-12x^3+x^4}$
3	$\frac{6+6x+3x^2+x^3}{6}$	$\frac{24+18x+16x^2+x^3}{24-6x}$	$\frac{60+36x+9x^2+x^3}{60-24x+3x^2}$	$\frac{120+60x+12x^2+x^3}{120-60x+12x^2-x^3}$	$\frac{840+360x+60x^2+4x^3}{840-480x+120x^2-16x^3+x^4}$
4	$\frac{24+24x+12x^2+4x^3+x^4}{24}$	$\frac{120+96x+36x^2+8x^3+x^4}{120-24x}$	$\frac{360+240x+72x^2+12x^3+x^4}{360-120x+12x^2}$	$\frac{840+480x+120x^2+16x^3+x^4}{840-360x+60x^2-4x^3}$	$\frac{1680+840x+180x^2+20x^3+x^4}{1680-840x+180x^2-20x^3+x^4}$

在表中令 $x=1$，可得 e 的相应帕德逼近，

$[1/1]=3$，

$[2/2]=19/7\approx 2.71428571$，

$[3/3]=193/71\approx 2.71830986$，

$[4/4]=2721/1001\approx 2.71828172$.

$[4/4]$的逼近与 e 的真值误差仅在第 7 位小数上相差 1. 如用连分式计算只用 4 次乘除法，相当于四阶的多项式逼近，但精度却高得多. 大量算例表明，在 $N=L+M$ 为一确定常数时，各种可能的$[L/M]$帕德逼近中，以 L 和 M 相等或接近相等者为最精确. 如，当 $N=2n$ 时，应采用$[n/n]$帕德逼近；当 $N=2n+1$ 时，应采用$[n+1/n]$或$[n/n+1]$帕德逼近. 总之，应采用帕德逼近表的主对角或主对角线附近的帕德逼近.

对于$[n/n]$帕德逼近，满足式(3.12.4)的方程可改为

$$p_0=a_0,\ p_j=\sum_{i=0}^{j}q_i a_{j-i}\quad (j=1,2,\cdots,n),\qquad (3.12.5)$$

其中 $q_0=1$，q_j 是下列方程的解：

$$\begin{cases} a_1q_n+a_2q_{n-1}+\cdots+a_nq_1+a_{n+1}=0,\\ a_2q_n+a_3q_{n-1}+\cdots+a_{n+1}q_1+a_{n+2}=0,\\ \vdots\\ a_nq_n+a_{n+1}q_{n-1}+\cdots+a_{2n-1}q_1+a_{2n}=0. \end{cases}\qquad (3.12.6)$$

将它写成矩阵形式为

$$\boldsymbol{A}_n\boldsymbol{q}+\boldsymbol{b}=\boldsymbol{0},$$

其中

$$\boldsymbol{A}_n=\begin{pmatrix} a_1 & a_2 & \cdots & a_n\\ a_2 & a_3 & \cdots & a_{n+1}\\ \vdots & \vdots & & \vdots\\ a_n & a_{n+1} & \cdots & a_{2n-1} \end{pmatrix},\quad \boldsymbol{q}=\begin{pmatrix} q_n\\ q_{n-1}\\ \vdots\\ q_1 \end{pmatrix},\quad \boldsymbol{b}=\begin{pmatrix} a_{n+1}\\ a_{n+2}\\ \vdots\\ a_{2n} \end{pmatrix}.$$

$\boldsymbol{A}_n$ 是对称阵，若 $\boldsymbol{A}_n$ 非奇异，则方程(3.12.6)有解，将它的解

$q_1, \cdots, q_n$代入式(3.12.5)则得 $p_0, p_1, \cdots, p_n$. 于是可得[n/n]帕德逼近 $R_{nn}(x)$.

为了估计 $f(x)$与 $R_{nn}(x)$的误差,由式(3.12.3)可得

$$Q_n(x)\sum_{k=0}^{\infty}a_kx^k - p_n(x) = x^{2n+1}\sum_{k=0}^{\infty}r_kx^k,$$

其中 r_k 可通过比较 x 高于 x^{2n+1}次幂的系数得到:

$$r_k = \sum_{i=0}^{n}q_ia_{2n+k+1-i} \quad (k = 0,1,\cdots). \qquad (3.12.7)$$

通常$|a_k|$是减少的,因此当$|x|\leqslant 1$ 时,误差

$$\begin{aligned} E_n &= | f(x) - R_{nn}(x) | \\ &= \left|\frac{x^{2n+1}}{Q_n(x)}\sum_{k=0}^{\infty}r_kx^k\right| \approx | r_0 | | x^{2n+1} | \leqslant | r_0 |. \end{aligned}$$

由式(3.12.7)可得

$$r_0 = a_{n+1}q_n + a_{n+2}q_{n-1} + \cdots + a_{2n}q_1 + a_{2n+1}.$$

将此方程与方程组(3.12.6)联立,消去 $q_1, \cdots, q_n$ 就得到 r_0,或由克拉默法则得

$$q_0 = 1 = \frac{\det\begin{pmatrix} a_1 & a_2 & \cdots & a_n & 0 \\ a_2 & a_3 & \cdots & a_{n+1} & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ a_{n+1} & a_{n+2} & \cdots & a_{2n} & r_0 \end{pmatrix}}{\det\boldsymbol{A}_{n+1}} = r_0\frac{\det\boldsymbol{A}_n}{\det\boldsymbol{A}_{n+1}}.$$

记 $\Delta_n = \det\boldsymbol{A}_n$,由此求得

$$r_0 = \frac{\Delta_{n+1}}{\Delta_n}.$$

在求[n/n]帕德逼近之前,为了先确定 n,可先估计误差

$$E_n \approx | r_0 | = \left|\frac{\Delta_{n+1}}{\Delta_n}\right| \quad (当\ | x | \leqslant 1),$$

当 E_n 达到精度要求再计算[n/n]帕德逼近.

例 3.12.2 $f(x) = \cos x$,令 $x^2 = z$,考虑$|x|\leqslant 1$,

$$\cos x = \cos\sqrt{z} = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k)!} z^k,$$

取 $L=M=3$,计算 Δ_3,Δ_4.

解 因为 $a_k=\frac{(-1)^k}{(2k)!}$,故

$$r_0 = \frac{\Delta_4}{\Delta_3} = \frac{45469}{59 \times 660 \times 14!} < 1.34 \times 10^{-11}.$$

当$|x|\leqslant 1$,则 $E_n<1.34\times 10^{-11}$.若$|x|\leqslant\frac{\pi}{3}=1.048$,而$(1.048)^{14}\leqslant 2$,故 $E_n\leqslant r_0(1.048)^{14}<2.68\times 10^{-11}$,仍能达到 10 位有效数字.由式(3.12.6)可求得

$$q_1 = 0.0294437956,$$
$$q_2 = 0.000425456685,$$
$$q_3 = 0.00000326957732.$$

由式(3.12.5)可求得

$$p_0 = 1,$$
$$p_1 = -0.4705562044,$$
$$p_2 = 0.0273702256,$$
$$p_3 = -0.0003715227.$$

于是得到

$$\cos x \approx \frac{1 + p_1 x^2 + p_2 x^4 + p_3 x^6}{1 + q_1 x^2 + q_2 x^4 + q_3 x^6}.$$

3.13 梅利逼近

帕德逼近是以函数的幂级数展开为基础的,其误差近似为 $r_0 x^{L+M+1}$,当$|x|\ll 1$ 时误差很小,但当$|x|$增大时误差迅速增大,特别当函数的幂级数展开收敛较慢时,效果较差.梅利(Maehly)逼近是帕德逼近的改进,它利用 $f(\cdot)$的切比雪夫展开求函数的

有理逼近，使逼近精度大大提高.

设 $f(x)$ 在 $[-1,1]$ 上的切比雪夫展开式为

$$f(x)=\sum_{k=0}^{\infty}c_k\mathrm{T}_k(x)\quad(|x|\leqslant 1),\qquad(3.13.1)$$

其中 $\mathrm{T}_k(x)=\cos(k\arccos x)$ 为切比雪夫多项式，级数(3.13.1)是一致收敛和绝对收敛的，在 $[-1,1]$ 上用 $R_{mn}(x)$ 做有理逼近，有

$$\begin{aligned}R_{mn}(x)&=\frac{p_m(x)}{Q_n(x)}\\&=\frac{a_0\mathrm{T}_0(x)+a_1\mathrm{T}_1(x)+\cdots+a_m\mathrm{T}_m(x)}{b_0\mathrm{T}_0(x)+b_1\mathrm{T}_1(x)+\cdots+b_n\mathrm{T}_n(x)},\end{aligned}\qquad(3.13.2)$$

这里系数 $a_0,\cdots,a_m$ 及 $b_0,\cdots,b_n$ 是根据帕德逼近的思想由下列 $m+n+1$ 个方程确定的非零解：

$$\begin{cases}a_0=b_0c_0+\dfrac{1}{2}\displaystyle\sum_{r=1}^{n}b_rc_r,\\ a_k=b_0c_k+\dfrac{1}{2}b_kc_0+\dfrac{1}{2}\displaystyle\sum_{r=1}^{n}b_r(c_{r+k}+c_{|r-k|}),\end{cases}\quad k=1,\cdots,m+n,$$

$$(3.13.3)$$

其中为书写方便，假定了当 $k>m$ 时取 $a_k=0$，当 $k>n$ 时，取 $b_k=0$. 由于方程组(3.13.3)中未知数个数多于方程个数，故非零解一定存在. 利用切比雪夫多项式的恒等性质

$$\mathrm{T}_r(x)\mathrm{T}_k(x)=\frac{1}{2}(\mathrm{T}_{r+k}(x)+\mathrm{T}_{|r-k|}(x)),$$

可以证明满足方程(3.13.3)的任一组解 $a_0,\cdots,a_m$ 及 $b_0,\cdots,b_n$ 所得的切比雪夫级数 $P_m(x)-Q_n(x)f(x)$ 的前 $m+n+1$ 项系数为 0. 记

$$P_m(x)-Q_n(x)f(x)=\sum_{k=m+n+1}^{\infty}h_k\mathrm{T}_k(x),$$

则

$$h_{m+n+1}=c_{m+n+1}+\frac{1}{2}\sum_{i=1}^{n}b_i(c_{m+n+i+1}+c_{m+n-i+1}),\qquad(3.13.4)$$

误差函数为

$$E_n(x) = R_{mn}(x) - f(x) = \frac{1}{Q_n(x)} \sum_{k=m+n+1}^{\infty} h_k \mathrm{T}_k(x).$$

通常，由于取 $b_0=1$，量 h_k 下降很快，所以

$$E_n(x) \approx h_{m+n+1} \mathrm{T}_{m+n+1}(x),$$

因而 $R_{mn}(x)$几乎是 f 在$[-1,1]$上的最佳有理逼近.

实际计算时，常取 $m=n, b_0=1$，这时式(3.13.3)变成

$$\begin{cases} a_0 = c_0 + \dfrac{1}{2}\displaystyle\sum_{r=1}^{n} b_r c_r, \\ a_k = c_k + \dfrac{1}{2} b_k c_0 + \dfrac{1}{2}\displaystyle\sum_{r=1}^{n} b_r (c_{r+k} + c_{|r-k|}), \end{cases} \quad k = 1, \cdots, n. \tag{3.13.5}$$

$$\frac{1}{2}\sum_{r=1}^{n}(c_{n+k+r} + c_{n+k-r}) b_r + c_{n+k} = 0 \quad (k = 1, \cdots, n). \tag{3.13.6}$$

由式(3.13.6)求出解 $b_1, \cdots, b_n$，代入式(3.13.5)可求出 $a_0, \cdots, a_n$. 为了估计误差，由式(3.13.4)有

$$h_{2n+1} = c_{2n+1} + \frac{1}{2}\sum_{i=1}^{n} b_i (c_{2n+i+1} + c_{2n-i+1}).$$

将此式与式(3.13.6)联立，应用克拉默法则可求得 $b_0 = \dfrac{h_{2n+1}\Delta_n}{D_n} = 1$，

即 $h_{2n+1} = \dfrac{D_n}{\Delta_n}$，其中

$$D_n = \begin{vmatrix} \frac{1}{2}(c_1 + c_{2n+1}) & \frac{1}{2}(c_2 + c_{2n}) & \cdots & \frac{1}{2}(c_n + c_{n+2}) & c_{n+1} \\ \frac{1}{2}(c_2 + c_{2n+2}) & \frac{1}{2}(c_3 + c_{2n+1}) & \cdots & \frac{1}{2}(c_{n+1} + c_{n+3}). & c_{n+2} \\ \vdots & \vdots & & \vdots & \vdots \\ \frac{1}{2}(c_{n+1} + c_{3n+1}) & \frac{1}{2}(c_{n+2} + c_{3n}) & \cdots & \frac{1}{2}(c_{2n} + c_{2n+2}). & c_{2n+1} \end{vmatrix},$$

Δ_n 是行列式 D_n 去掉最后一行及最后一列的代数余子式.

如果函数 f 具有奇偶性,为提高逼近精度可作变换 $z=x^2$,若 f 为偶函数,这时

$$f(x)=\sum_{k=0}^{\infty}c_k\mathrm{T}_{2k}(x)\quad(|x|\leqslant 1),$$

则

$$R_{mn}(x)=\frac{\sum_{k=0}^{m}a_k\mathrm{T}_{2k}(x)}{\sum_{k=0}^{n}b_k\mathrm{T}_{2k}(x)},$$

其中系数 a_k 及 b_k 可用式(3.13.3)的方法确定;当 f 为奇函数时

$$f(x)=\sum_{k=0}^{\infty}c_k\mathrm{T}_{2k+1}(x)$$

可改写成

$$\frac{f(x)}{x}=\frac{1}{2}c_0^*+\sum_{k=1}^{\infty}c_k^*\mathrm{T}_{2k}(x),$$

其中

$$c_k^*=2\sum_{j=0}^{\infty}(-1)^jc_{k+j},\quad 2c_k=c_k^*+c_{k+1}^*,$$

然后用梅利方法求 $x^{-1}f(x)$ 的有理逼近 $R_{mn}(x)$.

例 3.13.1 求 $f(x)=\arctan x$ 在 $|x|\leqslant 1$ 时的梅利逼近.

解 由于 $\arctan x$ 为奇函数,且

$$\arctan x=2\sum_{k=0}^{\infty}(-1)^k\frac{(\sqrt{2}-1)^{2k+1}}{2k+1}\mathrm{T}_{2k+1}(x)\quad(|x|\leqslant 1),$$

于是

$$x^{-1}\arctan x=\frac{1}{2}c_0^*+\sum_{k=1}^{\infty}c_k^*\mathrm{T}_{2k}(x),$$

其中

$$\frac{1}{2}c_0^* = \sum_{j=0}^{\infty}(-1)^j c_j = 2\sum_{j=0}^{\infty}\frac{(\sqrt{2}-1)^{2j+1}}{2j+1}$$

$$= \ln(\sqrt{2}+1) = 0.881373587019543.$$

再由递推公式

$$c_{k+1}^* = 2c_k - c_k^* \quad (k = 0,1,\cdots),$$

可以算出

$$c_1^* = -0.105892924546706, \quad c_2^* = 0.011135842059403,$$

$$c_3^* = -0.001381195003598, \quad c_4^* = 0.000185742973276,$$

$$c_5^* = -0.000026215196110, \quad c_6^* = 0.000003821036590,$$

$$c_7^* = -0.000000569918604.$$

当 $m=n=2$ 时,由方程(3.13.6)得

$$\begin{cases}(c_2^* + c_4^*)b_1 + (c_1^* + c_5^*)b_2 = -2c_3^*, \\ (c_3^* + c_5^*)b_1 + (c_2^* + c_6^*)b_2 = -2c_4^*.\end{cases}$$

解此方程得

$$b_1 = 0.37360540753, \quad b_2 = 0.01385410972.$$

代入式(3.13.5)得

$$a_0 = 0.8616696410, \quad a_1 = 0.2247301252,$$

$$a_2 = 0.0033086795.$$

于是

$$\arctan x \approx \frac{x[a_0 + a_1 T_2(x) + a_2 T_4(x)]}{1 + b_1 T_2(x) + b_2 T_4(x)}$$

$$= \frac{x(\alpha_0 + \alpha_1 x^2 + \alpha_2 x^4)}{\beta_0 + \beta_1 x^2 + \beta_2 x^4},$$

其中

$\alpha_0 = 0.6402481953$, $\alpha_1 = 0.4229908144$, $\alpha_2 = 0.0264694361$,

$\beta_0 = 0.6402487022$, $\beta_1 = 0.6363779373$, $\beta_2 = 0.0108328778$.

由式(3.13.4)可得

$$h_5 = c_5^* + \frac{1}{2}(c_4^* + c_6^*)b_1 + \frac{1}{2}(c_3^* + c_7^*)b_2 \approx 3 \times 10^{-7},$$

于是误差估计为

$$E_5(x) \approx h_5 x \mathrm{T}_{10}(x).$$

3.14 函数的连分式展开

函数的连分式展开也是获得函数有理逼近的一种简单方法. 同幂级数展开一样, 如果 $f(\cdot)$ 在原点解析, 则函数可展成连分式, 展开方法可以利用 $f(\cdot)$ 的幂级数展开式推出, 也可利用 $f(\cdot)$ 满足的常微分方程求其连分式解得到.

利用幂级数展开常用方法如下:

$$\begin{aligned}\frac{p_0 + p_1 x + p_2 x^2 + \cdots}{q_0 + q_1 x + q_2 x^2 + \cdots} &= \frac{p_0}{q_0} + \frac{p_0 + p_1 x + p_2 x^2 + \cdots}{q_0 + q_1 x + q_2 x^2 + \cdots} - \frac{p_0}{q_0} \\ &= \frac{p_0}{q_0} + \frac{x(p'_0 + p'_1 x + p'_2 x^2 + \cdots)}{q_0 + q_1 x + q_2 x^2 + \cdots} \\ &= \frac{p_0}{q_0} + \cfrac{x}{\cfrac{q_0 + q_1 x + q_2 x^2 + \cdots}{p'_0 + p'_1 x + p'_2 x^2 + \cdots}}.\end{aligned}$$

例 3.14.1 $\mathrm{e}^x = 1 + x + \frac{1}{2!}x^2 + \frac{1}{3!}x^3 + \cdots$

$$= \cfrac{1}{1 + \cfrac{1}{1 + x + \frac{1}{2!}x^2 + \cdots} - 1}$$

$$= \cfrac{1}{1 - \cfrac{x(1 + x/2! + x^2/3! + \cdots)}{1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots}}$$

$$=\cfrac{1}{1-\cfrac{x}{1+\cfrac{1+x+x^2/2!+x^3/3!+\cdots}{1+\frac{x}{2!}+\frac{x^2}{3!}+\cdots}-1}}$$

$$=\cfrac{1}{1-\cfrac{x}{1+\cfrac{x(1/2+x/3+x^2/8+\cdots)}{1+\frac{1}{2}x+\frac{1}{6}x^2+\cdots}}}$$

$$=\cfrac{1}{1-\cfrac{x}{1+\cfrac{x}{2+\cfrac{1+\frac{x}{2}+\frac{x^2}{6}+\cdots}{\frac{1}{2}+\frac{x}{3}+\frac{x^2}{8}+\cdots}-2}}}=\cdots$$

例 3.14.2 $\tan x=\dfrac{x-\frac{x^3}{3!}+\frac{x^5}{5!}-\cdots}{1-\frac{x^2}{2!}+\frac{x^4}{4!}-\cdots}$

$$=\cfrac{x}{1+\cfrac{1-\frac{x^2}{2!}+\frac{x^4}{4!}-\cdots}{1-\frac{x^2}{3!}+\frac{x^4}{5!}-\cdots}-1}$$

$$=\cfrac{x}{1+\cfrac{-x^2\left(\frac{1}{3}-\frac{x^2}{30}+\cdots\right)}{1-\frac{x^2}{3!}+\frac{x^4}{5!}-\cdots}}$$

$$=\cfrac{x}{1-\cfrac{x^2}{3+\cfrac{1-x^2/3!+x^4/5!-\cdots}{\frac{1}{3}-\frac{x^2}{30}+\frac{x^4}{7.5!}-\cdots}-3}}$$

$$=\cfrac{x}{1-\cfrac{x^2}{3-\cfrac{x^2}{5+\cfrac{1-x^2/10+\cdots}{\frac{1}{5}-\frac{x^2}{70}+\cdots}-5}}}$$

$$=\cdots$$

利用幂级数求函数的连分式展开通常只能得到前若干项系数,要得到 $f(x)$ 连分式展开的一般项可用常微分方程连分式解法.设有微分方程

$$y'=f(x,y),\quad y(0)=0, \tag{3.14.1}$$

当 x 很小时可求得 $y_0\approx\xi_0(x)$,令 $y=\frac{\xi_0(x)}{1+y_1}$为方程(3.14.1)的解. y_1 适合微分方程 $y'_1=f_1(x_1,y_1)$,$y_1(0)=0$,它的解是 $y_1=\frac{\xi_1(x)}{1+y_2}$,重复这一步骤可得方程(3.14.1)的连分式解为

$$y=\frac{\xi_0(x)}{1}+\frac{\xi_1(x)}{1}+\cdots+\frac{\xi_n(x)}{1}+\cdots.$$

为使 $\xi_v(x)$计算有规律,通常要使 f 与 f_1 有相同形式,这样 $\xi_0(x)$ 与 $\xi_1(x)$也有相同形式,并可通过逐次递推得到,通常取 $\xi_v(x)=a_vxk_v$ 的形式,其中 $k_v\geqslant0$.

方程

$$\begin{cases}(1+\eta x^k)\dfrac{xy'}{k}+(\beta+\alpha x^k)y+\gamma y^2=\delta x^k,\\ y(0)=0,\end{cases} \tag{3.14.2}$$

其中 $k,\eta,\beta,\alpha,\gamma,\delta$ 均为常数,且 $k>0$.这是一种特殊的 Riccati 方程,很多基本初等函数的连分式展开可利用这个方程的解得到.用连分式求方程(3.14.2)的解,在$|x|\ll1$ 时,方程高阶项可忽略不计,将方程(3.14.2)简化为线性方程

$$\frac{xy'_0}{k}+\beta y_0=\delta x^k,\quad y_0(0)=0,$$

此方程的解为

$$y_0(x)=\frac{\delta x^k}{1+\beta}=\xi_0(x).$$

令方程(3.14.2)的解为

$$y(x)=\frac{\delta x^k}{1+\beta+y_1},$$

则

$$y'=\frac{k\delta x^{k-1}(1+\beta+y_1)-\delta x^k y'_1}{(1+\beta+y_1)^2}.$$

将它代入方程(3.14.2)中,化简,并令 $\beta_1=1+\beta,\alpha_1=-(\eta+\alpha)$, $\gamma_1=1$, $\delta_1=\gamma\delta+(1+\beta)(\eta+\alpha)$,可得到

$$(1+\eta x^k)\frac{xy'_1}{k}+(\beta_1+\alpha_1 x^k)y_1+\gamma_1 y_1^2=\delta_1 x^k,$$

$$y_1(0)=0. \tag{3.14.3}$$

与解方程(3.14.2)类似,可得方程(3.14.3)的解为

$$y_1=\frac{\delta_1 x^k}{1+\beta_1+y_2},$$

代入式(3.14.3)得到关于 y_2 的微分方程

$$(1+\eta x^k)\frac{xy'_2}{k}+(\beta_2+\alpha_2 x^k)y_2+\gamma_2 y_2^2=\delta_2 x^k,y_2(0)=0,$$

其中 $\beta_2=1+\beta_1=2+\beta,\alpha_2=-(\eta+\alpha_1)=\alpha,\gamma_2=1,$

$$\delta_2=\gamma_1\delta_1+(1+\beta_1)(\eta+\alpha_1)=\gamma\delta+(1+\beta)\eta-\alpha.$$

依此类推,有

$$y_{n-1}=\frac{\delta_{n-1}x^k}{1+\beta_{n-1}+y_n},$$

$$(1+\eta x^k)\frac{xy'_n}{k}+(\beta_n+\alpha_n x^k)y_n+\gamma_n y_n^2=\delta_n x^k,$$

$$y_n(0)=0,$$

其中

$$\begin{cases}\beta_n = 1+\beta_{n-1}, \quad \alpha_n = -(\eta+\alpha_{n-1}), \quad \gamma_n = 1, \\ \delta_n = \gamma_{n-1}\delta_{n-1} + (1+\beta_{n-1})(\eta+\alpha_{n-1}), \quad n = 2,3,\cdots \end{cases} \tag{3.14.4}$$

于是方程(3.14.2)的连分式解为

$$y = \frac{\delta x^k}{1+\beta+}\frac{\delta_1 x^k}{1+\beta_1}+\cdots+\frac{\delta_n x^k}{1+\beta_n}+\cdots, \tag{3.14.5}$$

其中 $\beta_1,\beta_2,\cdots,\delta_1,\delta_2,\cdots$ 可由递推公式(3.14.4)得到

$$\beta_m = m+\beta \quad (m=1,2,\cdots),$$

$$\delta_m = \gamma_1\delta_1 + \sum_{j=1}^{m-1}(1+\beta_j)(\eta+\alpha_j) \quad (m \geqslant 2).$$

在 $m=2n$ 及 $m=2n+1$ 的情况下可得到

$$\delta_{2n} = \gamma\delta + (n+\beta)n\eta - n\alpha \quad (n=1,2,\cdots),$$

$$\delta_{2n+1} = \gamma\delta + (\beta+n+1)[(n+1)\eta+\alpha] \quad (n=0,1,\cdots).$$

将以上结果代入式(3.14.5)就得到方程(3.14.2)的解

$$y = \frac{\delta x^k}{1+\beta+}\frac{[\gamma\delta+(1+\beta)(\eta+\alpha)]x^k}{2+\beta}+\frac{[\gamma\delta+(1+\beta)\eta-\alpha]x^k}{3+\beta}+\cdots+\frac{[\gamma\delta+(\beta+n)(n\eta+\alpha)]x^k}{2n+\beta}+\frac{[\gamma\delta+(n+\beta)n\eta-n\alpha]x^k}{2n+1+\beta}+\cdots. \tag{3.14.6}$$

对于函数 $y=\tan x$,满足微分方程

$$y' = 1+y^2, \quad y(0)=0. \tag{3.14.7}$$

令 $y=\frac{x}{1+z}$,$z(0)=0$,将它代入方程(3.14.7)得到关于 z 的微分方程

$$\frac{xz'}{2}+\frac{1}{2}z+\frac{1}{2}z^2 = -\frac{1}{2}x^2, \quad z(0)=0. \tag{3.14.8}$$

与方程(3.14.2)比较得

$$\eta=0,\quad k=2,\quad \beta=\gamma=\frac{1}{2},\quad \alpha=0,\quad \delta=-\frac{1}{2},$$

故利用式(3.14.6)可得(3.14.8)的连分式解为

$$z=\frac{-\frac{1}{2}x^2}{3/2}+\frac{-\frac{1}{4}x^2}{5/2}+\frac{-\frac{1}{4}x^2}{7/2}+\cdots+\frac{-\frac{1}{4}x_2}{\frac{2n+1}{2}}+\cdots$$

$$=\frac{-x_2}{3}-\frac{x^2}{5}-\frac{x^2}{7}-\cdots-\frac{x^2}{2n+1}+\cdots. \tag{3.14.9}$$

于是,$y=\tan x$ 的连分式展开为

$$\tan x=\frac{x}{1}-\frac{x^2}{3}-\frac{x^2}{5}-\frac{x^2}{7}-\cdots-\frac{x^2}{2n+1}-\cdots. \tag{3.14.10}$$

由 $\tanh x=\mathrm{i}\tan\frac{x}{\mathrm{i}}$,用$\frac{x}{\mathrm{i}}$代替上式的 x 并乘 i 得

$$\tanh x=\frac{x}{1}+\frac{x^2}{3}+\frac{x^2}{5}+\cdots+\frac{x^2}{2n+1}+\cdots.$$

若考虑 $y=\arctan x$,满足微分方程

$$y'=\frac{1}{1+x^2},\quad y(0)=0.$$

令 $y=\frac{x}{1+z}$,$z(0)=0$,可得

$$(1+x^2)\frac{xz'}{2}+\left(\frac{1}{2}-\frac{1}{2}x^2\right)z+\frac{1}{2}z^2=\frac{1}{2}x^2,$$

$$z(0)=0. \tag{3.14.11}$$

与方程(3.14.2)比较得

$$\eta=1,\quad k=2,\quad \beta=\gamma=\delta=\frac{1}{2},\quad \alpha=-\frac{1}{2}.$$

由式(3.14.6)可得连分式解

$$
z=\frac{x^2/2}{3/2}+\frac{\left(\frac{1}{4}+\frac{3}{4}\right)x^2}{5/2}+\frac{\left(\frac{1}{4}+2\right)x^2}{7/2}+\cdots+
$$

$$
\frac{\left[\frac{1}{4}+\frac{(2n+1)(2n-1)}{4}\right]x^2}{\frac{4n+1}{2}}+\frac{\left[\frac{1}{4}+n^2+n\right]x^2}{\frac{4n+3}{2}}+\cdots
$$

$$
=\frac{x^2}{3}+\frac{4x^2}{5}+\frac{9x^2}{7}+\cdots+\frac{4n^2x^2}{4n+1}+\frac{(2n+1)^2x^2}{4n+3}+\cdots,
$$

于是得

$$
\arctan x=\frac{x}{1}+\frac{x^2}{3}+\frac{4x^2}{5}+\frac{9x^2}{7}+\cdots+\frac{n^2x^2}{2n+1}+\cdots.
$$

由于$\frac{1}{2}\ln\frac{1+x}{1-x}=\operatorname{arctanh}x=\mathrm{i}\arctan\frac{x}{\mathrm{i}}$,所以有

$$
\frac{1}{2}\ln\frac{1+x}{1-x}=\frac{x}{1}-\frac{x^2}{3}-\frac{4x^2}{5}-\cdots-\frac{n^2x^2}{2n+1}-\cdots.
$$

应用方程(3.14.2)的连分式解(3.14.6)还可得到下列基本初等函数的连分式展开:

$$
(1+x)^v=\frac{1}{1}-\frac{vx}{1}+\frac{(1+v)x}{2}+\frac{(1-v)x}{3}+
$$

$$
\frac{(2+v)x}{2}+\frac{(2-vx)}{5}+\cdots+
$$

$$
\frac{(n+v)x}{2}+\frac{(n-v)x}{2n+1}+\cdots;
$$

$$
\ln(1+x)=\frac{x}{1}+\frac{x}{2}+\frac{x}{3}+\frac{2x}{2}+\frac{2x}{5}+\cdots+
$$

$$
\frac{nx}{2}+\frac{nx}{2n+1}+\cdots;
$$

$$
\mathrm{e}^x=\frac{1}{1}-\frac{x}{1}+\frac{x}{2}-\frac{x}{3}+\frac{x}{2}-\frac{x}{5}+\cdots+
$$

$$
\frac{x}{2}-\frac{x}{2n+1}-\cdots
$$

也是近似的.

研究数值积分的另一个原因是被积函数没有具体的解析表达式,如实验数据等.因此无法采用解析方法求解定积分,只能用近似求积方法.同样地,由实验数据构成的表格形成的函数,需求其导数时也只能用数值方法来近似求解.

4.2 牛顿-科茨求积公式

4.2.1 公式的一般形式

将积分(4.1.1)中的积分区间$[a,b]$分成n等分,其节点x_k为

$$x_k = a + kh,\quad h = \frac{1}{n}(b-a)\quad (k = 0,1,\cdots,n).$$

对于给定的函数f,在节点$x_k(k=0,1,\cdots,n)$上的值$f(x_k)$为已知.那么f在$n+1$个节点$x_0,x_1,\cdots,x_n$上的n次代数插值多项式为

$$P_n(x) = \sum_{k=0}^{n}\left(\prod_{\substack{j=0\\ j\neq k}}^{n}\frac{x-x_j}{x_k-x_j}\right)f(x_k).$$

如果记$x=a+th$,则上式可以写为

$$P_n(x) = \sum_{k=0}^{n}\left(\prod_{\substack{j=0\\ j\neq k}}^{n}\frac{t-j}{k-j}\right)f(x_k). \tag{4.2.1}$$

在积分(4.1.1)中的被积函数f用其$n+1$个节点的代数插值多项式P_n来代替,可得

$$I(f) = \int_a^b f(x)\mathrm{d}x \approx I_n(f) = \int_a^b P_n(x)\mathrm{d}x.$$

多项式的积分是容易求出的,因此把上式写为

$$I(f) \approx I_n(f) = \sum_{k=0}^{n} A_k f(x_k), \tag{4.2.2}$$

其中

4　数值积分与数值微分

4.1　引言

在科学与工程计算中，经常要计算定积分

$$I(f)=\int_a^b f(x)\mathrm{d}x \quad (-\infty\leqslant a\leqslant b\leqslant\infty). \tag{4.1.1}$$

这个积分的计算似乎很简单，只要求出 f 的原函数 F 就可以得出积分(4.1.1)的值，即

$$I(f)=F(b)-F(a). \tag{4.1.2}$$

如果原函数 F 非常简单又便于使用，那么式(4.1.2)就提供了计算起来最快的积分法. 但是，积分过程往往将导出新的超越函数，例如，简单积分 $\int\frac{1}{x}\mathrm{d}x$ 可引出对数函数，它已不是代数函数了；而积分 $\int \mathrm{e}^{-x^2}\mathrm{d}x$，将引出一个无法用有限个代数运算、对数运算或指数运算组合表示的函数. 有些积分虽然容易求解，并且原函数仍然是一个初等函数，但可能过于复杂，以致人们采用式(4.1.2)来计算之前还得三思而行. 例如

$$\int\frac{1}{x^4+1}\mathrm{d}x=\frac{1}{2\sqrt{2}}\arctan\left(\frac{x\sqrt{2}}{1-x^2}\right)+\frac{1}{4\sqrt{2}}\ln\frac{x^2+x\sqrt{2}+1}{x^2-x\sqrt{2}+1}+C, \tag{4.1.3}$$

采用式(4.1.3)这种“精确”表达式时，所需运算次数是个根本问题. 由式(4.1.3)看出，需计算对数和反正切，因而只能计算到一定的近似程度. 由此可以看出，这类表面上是“精确”的方法，实际上

$$A_k = \frac{b-a}{n}\int_0^n \prod_{\substack{j=0 \\ j\neq k}}^n \frac{t-j}{k-j}\mathrm{d}t = (b-a)c_k^{(n)}, \qquad (4.2.3)$$

$$c_k^{(n)} = \frac{(-1)^{n-k}}{k!(n-k)!n}\int_0^n \prod_{\substack{j=0 \\ j\neq k}}^n (t-j)\mathrm{d}t. \qquad (4.2.4)$$

公式(4.2.2)称为**牛顿-科茨(Newton-Cotes)求积公式**或称为**等距节点求积公式**，A_k 称为求积公式系数，$c_k^{(n)}$ 称为科茨求积系数.

牛顿-科茨求积公式的误差估计 $E_n(f)=I(f)-I_n(f)$，由下面定理给出

定理 4.2.1 (1) 如果 n 为偶数，$f^{(n+2)}$ 在$[a,b]$上连续，则有

$$E_n(f) = c_n h^{n+3} f^{(n+2)}(\eta), \quad \eta\in[a,b], \qquad (4.2.5)$$

其中

$$c_n = \frac{1}{(n+2)!}\int_0^n t^2(t-1)(t-2)\cdots(t-n)\mathrm{d}t.$$

(2) 如果 n 为奇数，$f^{(n+1)}$ 在$[a,b]$上连续，则有

$$E_n(f) = c_n h^{n+2} f^{(n+1)}(\eta), \quad \eta\in[a,b], \qquad (4.2.6)$$

其中

$$c_n = \frac{1}{(n+1)!}\int_0^n t(t-1)(t-2)\cdots(t-n)\mathrm{d}t.$$

从定理 4.2.1 可以看出，如果 f 是 n 次多项式，得 $f^{(n+1)}(x)=0$，所以 $E_n(f)=0$. 由此可知，对于次数不高于 n 的多项式来说，牛顿-科茨公式(4.2.2)是精确成立的.

近似式

$$\int_a^b f(x)\mathrm{d}x \approx \sum_{k=0}^n A_k f(x_k) \qquad (4.2.7)$$

称为**一般的求积公式**，其中 x_k 称为**求积节点**，A_k 称为**求积系数**，亦称**伴随节点 x_k 的权**(weight). A_k 仅仅与节点 x_k 的选取有关而不依赖于被积函数 $f(x)$ 的形式.

数值积分(numerical integration)是一个近似方法,因此对尽可能多的被积函数 f,要求数值积分能准确地作出计算,这就引出了代数精度的概念.

定义 4.2.2 如果求积公式(4.2.7)对所有次数不高于 n 的代数多项式等式精确成立,但存在 $n+1$ 次的代数多项式使等式不成立,则称求积公式(4.2.7)具有 n 次代数精度.

由定理 4.2.1 可知,牛顿-科茨求积公式(4.2.2)的代数精度至少是 n 次,而当 n 是偶数时,(4.2.2)的代数精度可达 $n+1$ 次.

4.2.2 梯形公式

在牛顿-科茨公式(4.2.2)中,取 $n=1$ 时 $c_0^{(1)}=c_1^{(1)}=\dfrac{1}{2}$,所以有

$$I(f)\approx I_1(f)=\frac{b-a}{2}[f(a)+f(b)]. \qquad (4.2.8)$$

公式(4.2.8)称为**梯形公式**(trapezoidal rule). 如果用连接$(a, f(a))$和 $f(b,(b))$的直线来逼近 f,并对这线性函数进行积分可得到 $I_1(f)$. 图 4.1 中 $I_1(f)$就是梯形 $AabB$ 的面积,因此用 $I_1(f)$来逼近 $I(f)$,就是用梯形面积来代替曲边梯形面积.

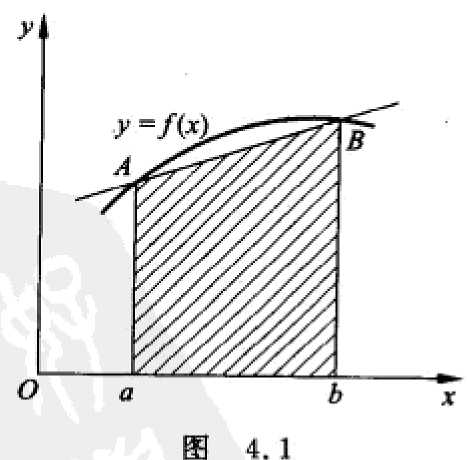


图 4.1

定理 4.2.3 若 $f\in C^2[a,b]$，则梯形公式(4.2.8)的误差为

$$E_1(f)=I(f)-I_1(f)=-\frac{1}{12}(b-a)^3f''(\eta),\quad \eta\in[a,b].$$

4.2.3 辛普森公式

在牛顿-科茨公式(4.2.2)中，取 $n=2$，则有

$$c_0^{(2)}=\frac{1}{4}\int_0^2(t-1)(t-2)\mathrm{d}t=\frac{1}{6},$$

$$c_1^{(2)}=-\frac{1}{2}\int_0^2 t(t-2)\mathrm{d}t=\frac{4}{6},$$

$$c_2^{(2)}=\frac{1}{4}\int_0^2 t(t-1)\mathrm{d}t=\frac{1}{6},$$

由此得到

$$\begin{aligned}I(f)&\approx I_2(f)\\&=\frac{h}{3}\left[f(a)+4f\left(\frac{a+b}{2}\right)+f(b)\right],\end{aligned}\qquad(4.2.9)$$

其中 $h=\frac{1}{2}(b-a)$. 式(4.2.9)称为**辛普森(Simpson)公式**，其几何意义就是用通过三点 $M_0(x_0,y_0)$，$M_1(x_1,y_1)$，$M_2(x_2,y_2)$ 的抛物线 P_2 围成的曲边形面积来代替给定函数 f 围成的曲边形面积(见图 4.2). 由于上述原因，公式(4.2.9)也称**抛物线公式**.

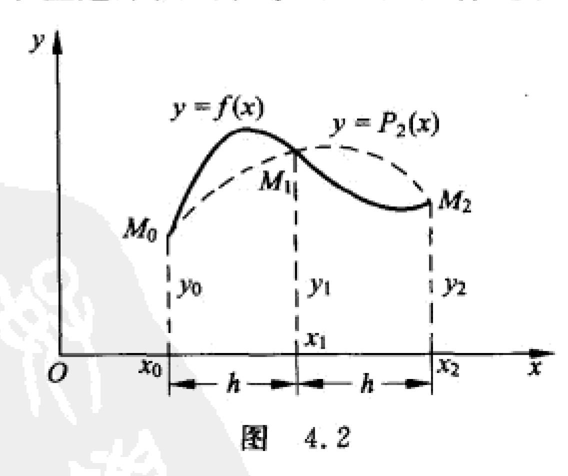


图 4.2

定理 4.2.4 若 $f\in C^4[a,b]$，则辛普森公式(4.2.9)的误差为

$$E_2(f)=I(f)-I_2(f)=-\frac{1}{90}h^5f^{(4)}(\eta),\quad \eta\in[a,b].$$

4.2.4 高阶牛顿-科茨公式

在牛顿-科茨公式(4.2.2)中，$n\geqslant 3$ 的公式称为**高阶牛顿-科茨公式**.

对于 $n=3,h=\frac{1}{3}(b-a)$，可以求得

$$\begin{aligned}I(f)&\approx I_3(f)\\&=\frac{3}{8}h[f(a)+3f(a+h)+3f(b-h)+f(b)].\end{aligned}\qquad(4.2.10)$$

此公式称为**辛普森 3/8 公式**. 如果 $f\in C^4[a,b]$，则公式(4.2.10)的误差为

$$\begin{aligned}E_3(f)&=I(f)-I_3(f)\\&=-\frac{3}{80}h^5f^{(4)}(\eta),\quad \eta\in[a,b].\end{aligned}\qquad(4.2.11)$$

在牛顿-科茨公式中，取 $n=4,h=\frac{1}{4}(b-a)$，则公式化为

$$\begin{aligned}I(f)&\approx I_4(f)\\&=\frac{2}{45}h\Big[7f(a)+32f(a+h)+\\&\quad 12f\Big(\frac{a+b}{2}\Big)+32f(b-h)+7f(b)\Big].\end{aligned}\qquad(4.2.12)$$

通常把这个公式称做**科茨公式**，也称**布尔(Boole)公式**. 如果 $f\in C^6[a,b]$，那么公式(4.2.12)的误差是

$$E_4(f)=-\frac{8}{945}h^7f^{(6)}(\eta),\quad \eta\in[a,b].\qquad(4.2.13)$$

在牛顿-科茨公式中，取 $n=5,h=\frac{1}{5}(b-a)$，则公式化为

$$I(f) \approx I_5(f)$$

$$= \frac{5h}{288}[19f(a) + 75f(a+h) + 50f(a+2h) +$$

$$50f(b-2h) + 75f(b-h) + 19f(b)]. \quad (4.2.14)$$

此公式称为牛顿-科茨六点公式.如果 $f \in C^6[a,b]$,那么公式(4.2.14)的误差是

$$E_5(f) = -\frac{275}{12096}h^7 f^{(6)}(\eta), \quad \eta \in [a,b]. \quad (4.2.15)$$

在牛顿-科茨公式中,取 $n=6, h=\frac{1}{6}(b-a)$,那么公式化为

$$I(f) \approx I_6(f)$$

$$= \frac{h}{140}[41f(a) + 216f(a+h) + 27f(a+2h) +$$

$$272f\left(\frac{a+b}{2}\right) + 27f(b-2h) + 216f(b-h) + 41f(b)].$$

$$(4.2.16)$$

此公式称为牛顿-科茨七点公式.若 $f \in C^8[a,b]$,那么公式(4.2.16)的误差是

$$E_6(f) = -\frac{9}{1400}h^9 f^{(8)}(\eta) \quad \eta \in [a,b]. \quad (4.2.17)$$

根据科茨系数公式(4.2.3),(4.2.4)有科茨系数表4.1($n \leqslant 8$).

表 4.1

n	$C_k^{(n)}$			
1	$\frac{1}{2}$	$\frac{1}{2}$		
2	$\frac{1}{6}$	$\frac{4}{6}$	$\frac{1}{6}$	
3	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{1}{8}$

续表

n	$C_k^{(n)}$								
4	$\frac{7}{90}$	$\frac{16}{45}$	$\frac{2}{15}$	$\frac{16}{45}$	$\frac{7}{90}$				
5	$\frac{19}{288}$	$\frac{25}{96}$	$\frac{25}{144}$	$\frac{25}{144}$	$\frac{25}{96}$	$\frac{19}{288}$			
6	$\frac{41}{840}$	$\frac{9}{35}$	$\frac{9}{280}$	$\frac{34}{105}$	$\frac{9}{280}$	$\frac{9}{35}$	$\frac{41}{840}$		
7	$\frac{751}{17280}$	$\frac{3577}{17280}$	$\frac{1323}{17280}$	$\frac{2989}{17280}$	$\frac{2989}{17280}$	$\frac{1323}{17280}$	$\frac{3577}{17280}$	$\frac{751}{17280}$	
8	$\frac{989}{28350}$	$\frac{5888}{28350}$	$\frac{-928}{28350}$	$\frac{10496}{28350}$	$\frac{-4540}{28350}$	$\frac{10496}{28350}$	$\frac{-928}{28350}$	$\frac{5888}{28350}$	$\frac{989}{28350}$

从误差分析来考虑，好像 n 越大，精度越高，但由于高阶牛顿-科茨求积公式的舍入误差的干扰比较大，因此，在实际计算中一般不宜采用 n 很大的牛顿-科茨公式. 特别当 $n \geqslant 8$ 时，牛顿-科茨公式中系数符号是混合的，计算中会丢失不少有效数字，所以不采用为宜.

4.2.5 开型牛顿-科茨公式

令 $h=(b-a)/(n+2)$，$x_0=a+h$，$x_k=x_0+kh\ (k=1,2,\cdots,n)$，于是 $x_n=b-h$. 用 $x_{-1}=a$，$x_{n+1}=b$ 表示积分区间$[a,b]$的端点，求积公式可写为

$$I(f)=\int_{x_{-1}}^{x_{n+1}} f(x)\mathrm{d}x \approx \sum_{k=0}^{n} A_k f(x_k), \qquad (4.2.18)$$

其中 $A_k(k=0,1,\cdots,n)$ 由公式(4.2.3)给出.

注意：在上述求积公式中，积分区间的端点皆没有采用.

如果在求积公式(4.2.2)中，积分区间的端点至少有一个不用，则称公式(4.2.2)为**开型牛顿-科茨求积公式**. 显然，公式(4.2.18)是开型牛顿-科茨公式，而公式(4.2.18)以前的牛顿-科茨公式称为**闭型公式**.

开型牛顿-科茨求积公式的误差有下面定理.

定理 4.2.5 设 $\sum_{k=0}^{n} A_k f(x_k)$ 表示 $n+1$ 个节点的开型牛顿-科茨求积公式,其中 $x_{-1}=a, x_{n+1}=b, h=(b-a)/(n+2)$.

(1) 如果 n 为偶数,$f^{(n+2)}$ 在 $[a,b]$ 上连续,则有

$$E_n(f)=c_n h^{n+3} f^{(n+2)}(\eta), \quad \eta\in[a,b], \tag{4.2.19}$$

其中

$$c_n=\frac{1}{(n+2)!}\int_{-1}^{n+1} t^2(t-1)\cdots(t-n)\mathrm{d}t.$$

(2) 如果 n 为奇数,$f^{(n+1)}$ 在 $[a,b]$ 上连续,则有

$$E_n(f)=c_n h^{n+2} f^{(n+1)}(\eta), \quad \eta\in[a,b], \tag{4.2.20}$$

其中

$$c_n=\frac{1}{(n+1)!}\int_{-1}^{n+1} t(t-1)\cdots(t-n)\mathrm{d}t.$$

在实际计算中,闭型公式一般要比同阶的开型公式更为精确,因此经常使用的是闭型牛顿-科茨求积公式.但在 $n=0$ 时,开型牛顿-科茨公式是经常使用的,其求积公式为

$$I(f)\approx I_0(f)=2hf(x_0), \tag{4.2.21}$$

其中

$$h=\frac{1}{2}(b-a), \quad x_0=\frac{1}{2}(a+b).$$

公式(4.2.21)称为**中点公式**(midpoint rule).如果 f 在 $[a,b]$ 上有二阶连续导数,则中点公式的误差为

$$E_0(f)=\frac{1}{3}h^3 f''(\eta), \quad \eta\in[a,b]. \tag{4.2.22}$$

其他常用开型牛顿-科茨公式如下:

$n=1$,

$$\int_{x_{-1}}^{x_2} f(x)\mathrm{d}x=\frac{3h}{2}[f(x_0)+f(x_1)]+\frac{3h^3}{4}f''(\xi), \tag{4.2.23}$$

其中 $x_{-1}<\xi<x_2$.

$n=2$,

$$\int_{x_{-1}}^{x_3} f(x)\mathrm{d}x = \frac{4h}{3}[2f(x_0) - f(x_1) + 2f(x_2)] + \frac{14h^5}{45} f^{(4)}(\xi), \tag{4.2.24}$$

其中 $x_{-1}<\xi<x_3$.

$n=3$,

$$\int_{x_{-1}}^{x_4} f(x)\mathrm{d}x = \frac{5h}{24}[11f(x_0) + f(x_1) + f(x_2) + 11f(x_3)] + \frac{95}{144}h^5 f^{(4)}(\xi), \tag{4.2.25}$$

其中 $x_{-1}<\xi<x_4$.

例 4.2.6 用闭型和开型牛顿-科茨公式(4.2.8),(4.2.9),(4.2.10),(4.2.12),(4.2.21),(4.2.23),(4.2.24)和(4.2.25)来计算积分 $\int_0^{\frac{\pi}{4}} \sin x \mathrm{d}x$ 的近似值.

解 积分解析解为 $1-\frac{\sqrt{2}}{2}\approx 0.29289322$. 计算结果见表 4.2.

表 4.2

n	0	1	2	3	4
闭型		0.27768018	0.29293264	0.29291070	0.29289318
误差		0.01521303	0.00003942	0.00001748	0.0000004
开型	0.30055887	0.29798754	0.29285866	0.29286923	
误差	0.00766565	0.00509432	0.00003456	0.00002399	

4.3 复合求积公式

对于定积分 $\int_1^{10} \frac{1}{x}\mathrm{d}x$,其精确值为 2.302585. 用梯形公式计算得 4.95,用辛普森公式计算得 2.740909. 可以看出,它们的误差很

大.要求得比较精确的积分值,必须用其他高阶求积公式,但是高阶($n\geqslant 8$)牛顿-科茨求积公式会使有效数字丢失,因此不能用提高牛顿-科茨求积公式的阶的办法来提高精度.基于这种原因,一般把整个积分区间分成若干个子区间(通常是等分),再在每个小区间上单独采用同一种低阶求积公式,这种方法称为**复合求积方法**(composite numerical integration).

将积分区间$[a,b]$分为n等分,步长$h=\frac{1}{n}(b-a)$,分点为$x_k=a+kh(k=0,1,\cdots,n)$.复化求积法的步骤是:先用低阶牛顿-科茨公式求得在每个子区间$[x_k,x_{k+1}]$上的积分近似值,然后对这些近似值求和,从而得到$I(f)$的近似值.

4.3.1 复合梯形公式

在每个小区间$[x_k,x_{k+1}]$上使用梯形求积公式,再求和得到逼近$I(f)$的求积公式

$$I(f)\approx T_n=\frac{1}{2}h\sum_{k=0}^{n-1}[f(x_k)+f(x_{k+1})]$$

$$=\frac{1}{2}h\Big[f(a)+2\sum_{k=1}^{n-1}f(x_k)+f(b)\Big]. \tag{4.3.1}$$

公式(4.3.1)称为**复合梯形公式**(composite trapezoidal rule),T_n的下标n表示积分区间$[a,b]$分成n等分.

对每个小区间$[x_k,x_{k+1}]$上的梯形公式进行误差估计,然后相加就能得到复合梯形公式的误差.

定理 4.3.1 设$f\in C^2[a,b]$,那么复合梯形公式(4.3.1)的误差为

$$E_n(f)=I(f)-T_n$$

$$=-\frac{(b-a)}{12}h^2f''(\eta),\quad \eta\in[a,b].$$

4.3.2 复合辛普森公式

类似于复合梯形公式,但在小区间$[x_k,x_{k+1}]$上不用梯形公式而采用辛普森公式,那么就可以得到逼近 $I(f)$的求积公式

$$
\begin{aligned}
I(f) &\approx S_n \\
&= \frac{h}{6}\sum_{k=0}^{n-1}\left[f(x_k)+4f(x_{k+\frac{1}{2}})+f(x_{k+1})\right], \qquad (4.3.2)
\end{aligned}
$$

其中 $x_{k+\frac{1}{2}}=\frac{1}{2}(x_k+x_{k+1})$. 公式(4.3.3)称为**复合辛普森公式**(composite Simpson's rule). S_n 的下标n 表示把积分区间$[a,b]$分为 n 等分. 若令

$$H_n = h\sum_{k=0}^{n-1}f(x_{k+\frac{1}{2}}),$$

那么可得复合辛普森公式和复合梯形公式之间的关系

$$S_n=\frac{1}{3}T_n+\frac{2}{3}H_n. \qquad (4.3.3)$$

定理 4.3.2 设 $f\in C^4[a,b]$,那么复合辛普森公式(4.3.2)的误差为

$$
\begin{aligned}
E_n(f) &= I(f)-S_n \\
&= -\frac{b-a}{2880}h^4 f^{(4)}(\eta), \quad \eta\in[a,b].
\end{aligned}
$$

例 4.3.3 分别用复合梯形公式和复合辛普森公式计算积分

$$I(f)=\int_0^{\pi}\mathrm{e}^x\cos x\mathrm{d}x.$$

解 积分 $I(f)$的精确值为

$$I(f)=-12.0703463164.$$

T_n,S_n 以及它们的误差 $E_n^{(t)}=I(f)-T_n$,$E_n^{(s)}=I(f)-S_n$ 见表 4.3.

表 4.3

n	T_n	$E_n^{(t)}$	S_n	$E_n^{(s)}$
2	−17.389259	5.32	−11.5928395534	-4.78×10^{-1}
4	−13.336023	1.27	−11.9849440198	-8.54×10^{-2}
8	−12.382162	3.12×10^{-1}	−12.0642089572	-6.14×10^{-3}
16	−12.148004	7.77×10^{-2}	−12.0699513233	-3.95×10^{-4}
32	−12.089742	1.94×10^{-2}	−12.0703214561	-2.49×10^{-5}
64	−12.075194	4.85×10^{-3}	−12.0703447599	-1.56×10^{-6}
128	−12.071558	1.21×10^{-3}	−12.0703462191	-9.73×10^{-8}
256	−12.070649	3.03×10^{-4}	−12.0703463103	-6.08×10^{-9}

复合辛普森公式计算的结果比复合梯形公式的计算结果精确得多,并随 n 的增加,误差减少得很快.

4.3.3 复合求积公式的收敛性

对于牛顿-科茨公式,当 $n\to\infty$时,一般不能保证 $I_n(f)\to I(f)$.

例 4.3.4 用牛顿-科茨求积公式计算积分

$$I(f)=\int_{-4}^{4}\frac{1}{1+x^2}\mathrm{d}x.$$

解 此积分精确值 $I(f)=2\arctan4\approx2.6516$. 牛顿-科茨求积公式 $I_n(f)$见表 4.4. 由表 4.4 的数值结果可以看出,$I_n(f)$是发散的.

表 4.4

n	$I_n(f)$
2	5.4902
4	2.2776
6	3.3288
8	1.9411
10	3.5956

对于复合求积公式来讲，情况大不相同．可以把复合梯形公式 T_n 和复合辛普森公式 S_n 看作积分和．因此只要 $f(x)$ 是黎曼(Riemann)可积的，那么 T_n 和 S_n 均收敛到 $I(f)$．

定义 4.3.5 设复合求积公式 $I_n \approx I(f)$，如果当 $h \to 0$ 时有

$$\frac{I(f) - I_n}{h^p} \to c,$$

其中 c 为一个非零常数，那么称 I_n 是 p 阶收敛的．

由定理 4.3.1 和定理 4.3.2 知，复合梯形公式是二阶收敛的，复合辛普森公式是四阶收敛的．

4.3.4 区间逐次分半法

利用复合梯形公式和复合辛普森公式来进行定积分的近似计算是比较简单的，但是为了确定把积分区间$[a,b]$分成多少个子区间，即 n 取多大，则需依据余项公式事先估计，就要分析被积函数的高阶导数，而这是很困难的．

区间逐次分半法就是根据规定的精度要求，在计算过程中把积分区间逐次分半，并利用前后两次计算结果来判别误差的大小，从而得到满足精度要求的积分近似值．

利用复合梯形公式的误差估计(定理 4.3.1)可以得到

$$\frac{1}{3}(T_{2n} - T_n) \approx I(f) - T_{2n}. \tag{4.3.4}$$

上式提供了一个误差估计的判别条件．如果

$$|T_{2n} - T_n| < \varepsilon \quad (允许误差),$$

那么可以认为 T_{2n} 已满足精度要求．

需要特别注意的是，在逐次分半过程中，老分点上的函数值要避免计算．

区间逐次分半法的具体计算过程如下：

最初取 $n=1$，计算

$$T_1 = h_0\left[\frac{f(a)}{2}+\frac{f(b)}{2}\right], \quad h_0 = b-a.$$

然后将区间分半，取 $n=2$，计算

$$T_2 = h_1\left[\frac{f(a)}{2}+\frac{f(b)}{2}+f(x_1)\right]$$
$$= \frac{1}{2}T_1 + h_1 f(x_1),$$

其中 $h_1=\frac{1}{2}(b-a)$，$\quad x_1=a+h_1$.

再将每个小区间分半，则 $n=4$，计算

$$T_4 = h_2\left[\frac{f(a)}{2}+\frac{f(b)}{2}+f(x_1)+f(x_2)+f(x_3)\right]$$
$$= \frac{1}{2}T_2 + h_2[f(x_1)+f(x_3)],$$

其中 $h_2=\frac{1}{4}(b-a)$，$x_k=a+kh_2 \quad (k=1,2,3)$.

一般地，每次总是在前一次的基础上再将小区间分半，则分点加密. 一般计算公式为

$$T_{2n} = \frac{1}{2}T_n + h_{2n}\sum_{k=1}^{n} f(a+(2k-1)h_{2n}), \qquad (4.3.5)$$

其中 $h_{2n}=\frac{1}{2n}(b-a)$.

利用公式(4.3.5)计算出 T_{2n}后，再检验不等式

$$|T_{2n}-T_n| < \epsilon \quad (\text{取绝对误差}) \qquad (4.3.6)$$

或

$$\frac{|T_{2n}-T_n|}{|T_{2n}|} < \epsilon \quad (\text{取相对误差}) \qquad (4.3.7)$$

是否满足，如果满足则取 T_{2n}为所求定积分的近似值，否则区间继续分半，重复上述过程直至条件满足. 一般在实际计算中，当 $n>n_0$ (n_0 为给定一正整数)时才进行条件(4.3.6)或条件(4.3.7)的判

别，否则可能出现假收敛.

对于辛普森求积公式也可以同样进行区间分半. 由复合辛普森求积公式的误差(定理 4.3.2)可以得到

$$\frac{1}{15}(S_{2n}-S_n)\approx I(f)-S_{2n}. \qquad (4.3.8)$$

因此可以由 $S_{2n}-S_n$ 来估计 S_{2n}的误差. 用区间逐次分半产生复合辛普森求积的序列

$$S_1, S_2, S_4, \cdots, S_n, S_{2n}, \cdots$$

其中

$$S_n=\frac{1}{3}h_n[f(a)+f(b)+2P_n+4Q_n] \quad (n=1,2,4,\cdots), \qquad (4.3.9)$$

P_n 是在旧分点上函数值之和，

$$P_n=\sum_{k=1}^{n-1}f(x_{2k}).$$

Q_n 是在新分点上函数值之和，

$$Q_n=\sum_{k=1}^{n}f(x_{2k-1}).$$

$$h_n=\frac{1}{2n}(b-a),$$

$$x_k=a+kh_n \quad (k=1,2,\cdots,2n-1).$$

在旧分点上的函数值不需重计算. 利用公式(4.3.9)计算出 S_{2n}之后，再进行精度检验. 如果满足条件

$$|S_{2n}-S_n|<\varepsilon \qquad (4.3.10)$$

或

$$\frac{|S_{2n}-S_n|}{|S_{2n}|}<\varepsilon, \qquad (4.3.11)$$

那么取 S_{2n}作为 $I(f)$的近似值，否则再分半区间继续进行.

4.4 里查森外推算法和龙贝格积分法

利用复合梯形求积公式来计算 $I(f)$ 的近似值，精度较差. 当 $n\to\infty$ 时，序列 $\{T_k, k\geqslant 1\}$ 收敛到 $I(f)$，但收敛速度缓慢，为加速其收敛速度，可以利用外推(extrapolation)算法来加速.

4.4.1 里查森外推算法

设函数 S 在 $x=0$ 处的值 $S(0)$ 由序列 $S(h), S\left(\frac{h}{2}\right), \cdots (h>0)$ 来逼近. 通过序列 $S(h), S\left(\frac{h}{2}\right), \cdots$ 构造出一个新的序列，使它更快地收敛于 $S(0)$. 运用泰勒展开式

$$S(h) = S(0) + hS'(0) + \frac{1}{2!}h^2 S''(0) + \frac{1}{3!}h^3 S'''(0) + \cdots$$

$$S\left(\frac{h}{2}\right) = S(0) + \frac{h}{2}S'(0) + \frac{1}{2!}\left(\frac{h}{2}\right)^2 S''(0) + \frac{1}{3!}\left(\frac{h}{2}\right)^3 S'''(0) + \cdots$$

如果 $S'(0)\neq 0$，那么 $S(h), S\left(\frac{h}{2}\right)$ 逼近 $S(0)$ 的阶为 $O(h)$.

若令 $S_1(h) = 2S\left(\frac{h}{2}\right) - S(h)$，那么当 $S''(0)\neq 0$ 时，$S_1(h)$ 逼近 $S(0)$ 的阶为 $O(h^2)$，因此序列 $S_1(h), S_1\left(\frac{h}{2}\right), \cdots$ 就更快地收敛到 $S(0)$. 同样，还可从 $S_1(h)$ 构造出 $S_2(h)$，从 $S_2(h)$ 构造出 $S_3(h)$，它们都加速了收敛. 这种加速收敛的办法称为**外推算法**. 在数值积分中常用的里查森(Richardson)外推算法如下：

设一个步长为 h 的函数 F 去逼近一个数 F^*，其误差为

$$E(F^*) = F^* - F(h)$$
$$= a_1 h^{p_1} + a_2 h^{p_2} + \cdots + a_k h^{p_k} + \cdots \quad (4.4.1)$$

其中

$$p_k > p_{k-1} > \cdots > p_2 > p_1 > 0,$$

a_i, p_i 是与 h 无关的常数,即 F 逼近 F^* 的误差阶是 h^{p_1}.如果令

$$F_2(h) = \frac{1}{1-q^{p_1}}[F(qh) - q^{p_1}F(h)], \quad 1-q^{p_1} \neq 0,$$

那么 $F_2(h)$逼近 F^* 的误差阶是 h^{p_2}.重复这样的做法,可以得到一个算法

$$\begin{cases} F_1(h) = F(h), \\ F_{m+1}(h) = \dfrac{1}{1-q^{p_m}}[F_m(qh) - q^{p_m}F_m(h)] \quad (m = 1,2,\cdots), \end{cases} \quad (4.4.2)$$

其中 q 为满足 $1-q^{p_m} \neq 0 (m=1,2,\cdots)$的适当正数.算法(4.4.2)称为**里查森外推算法**.

定理 4.4.1 如果 F 逼近 F^* 的截断误差由式(4.4.1)给出,那么算法(4.4.2)逼近 F^* 的误差为

$$F^* - F_{m+1}(h) = a_{m+1}^{(m+1)} h^{p_{m+1}} + a_{m+2}^{(m+1)} h^{p_{m+2}} + \cdots$$

其中 $a_k^{(m+1)} (k \geq m+1)$为与 h 无关的常数.

算法(4.4.2)的计算步骤见表 4.5,其中ⓘ表示第 i 步.

表 4.5

①$F(h)$				
②$F(qh)$	③$F_2(h)$			
④$F(q^2h)$	⑤$F_2(qh)$	⑥$F_3(h)$		
⑦$F(q^3h)$	⑧$F_2(q^2h)$	⑨$F_3(qh)$	⑩$F_4(h)$	
⋮	⋮	⋮	⋮	…

4.4.2 龙贝格积分法

定理 4.3.1 给出了复合梯形求积公式的误差，为利用里查森外推算法来加速收敛，可用复合梯形求积公式的另一形式的误差.

定理 4.4.2 设 $f\in C^{2m+2}[a,b]$，那么复合梯形求积公式(4.3.1)的误差为

$$\int_a^b f(x)\mathrm{d}x - T_n(f) = \sum_{l=1}^{m} \frac{\mathrm{B}_{2l}}{(2l)!}[f^{(2l-1)}(a) - f^{(2l-1)}(b)]h^{2l} + r_{m+1},$$

其中 B_{2l} 为伯努利(Bernoulli)数，表 4.6 给出了部分伯努利数.

$$r_{m+1} = -\frac{\mathrm{B}_{2m+2}}{(2m+2)!}(b-a)f^{(2m+2)}(\eta)h^{2m+2},\quad \eta\in[a,b].$$

表 4.6

k	B_k
0	$\frac{1}{1}=1.00000$
1	$-\frac{1}{2}=-0.50000$
2	$\frac{1}{6}\approx 0.16667$
4	$-\frac{1}{30}\approx -0.03333$
6	$\frac{1}{42}\approx 0.02381$
8	$-\frac{1}{30}\approx -0.03333$
10	$\frac{5}{66}\approx 0.07576$
12	$-\frac{691}{2730}\approx -0.25311$
14	$\frac{7}{6}\approx 1.16667$

续表

k	B_k
16	$-\frac{3617}{510}\approx-7.09216$
18	$\frac{43867}{798}\approx 54.97118$
20	$-\frac{174611}{330}\approx-529.12424$
22	$\frac{854513}{138}\approx 6192.12319$
24	$-\frac{236364091}{2730}\approx-86580.25311$

当 k 为大于等于 3 的奇数时，$B_k=0$

在复合梯形求积公式(4.3.1)中，用 $T(h)$ 来记 T_n，其中 h 为步长. 由此公式(4.3.1)可写成

$$I(f)\approx T(h)=\frac{h}{2}\Big[f(a)+2\sum_{k=1}^{n-1}f(a+kh)+f(b)\Big].$$

现将步长每次缩小一半，这样可得一序列

$$T(h),\quad T\left(\frac{h}{2}\right),\quad T\left(\frac{h}{2^2}\right),\cdots$$

显然，这序列收敛到积分值 $I(f)$. 利用定理 4.4.2 及理查森外推算法(4.4.2)，取 $q=\frac{1}{2}$ 可得到如下算法：

$$\begin{cases} T_1(h)=T(h), \\ T_{m+1}(h)=\dfrac{T_m\left(\dfrac{h}{2}\right)-\left(\dfrac{1}{2}\right)^{2m}T_m(h)}{1-\left(\dfrac{1}{2}\right)^{2m}} \end{cases}\quad (m=1,2,\cdots). \tag{4.4.3}$$

算法(4.4.3)称为**龙贝格(Romberg)积分法**. $T_{m+1}(h)$ 逼近 $I(f)$ 的误差估计为

$$I(f)-T_{m+1}(h)=a_{m+1}^{(m+1)}h^{2(m+1)}+a_{m+2}^{(m+1)}h^{2(m+2)}+\cdots \tag{4.4.4}$$

其中系数 $a_{m+1}^{(m+1)}, a_{m+2}^{(m+1)}, \cdots$ 皆与 h 无关.

令

$$T_m^{(k)}=T_m\left(\frac{b-a}{2^k}\right),$$

则计算过程可见表 4.7. 表中的每列与对角线都收敛到定积分 $I(f)$.

表 4.7

$T_1^{(0)}$						
$T_1^{(1)}$	$T_2^{(0)}$					
$T_1^{(2)}$	$T_2^{(1)}$	$T_3^{(0)}$				
$T_1^{(3)}$	$T_2^{(2)}$	$T_3^{(1)}$	$T_4^{(0)}$			
$T_1^{(4)}$	$T_2^{(3)}$	$T_3^{(2)}$	$T_4^{(1)}$	$T_5^{(0)}$		
$T_1^{(5)}$	$T_2^{(4)}$	$T_3^{(3)}$	$T_4^{(2)}$	$T_5^{(1)}$	$T_6^{(0)}$	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$

龙贝格积分法的计算步骤如下:

1. 计算

$$T_1^{(0)}=\frac{b-a}{2}[f(a)+f(b)].$$

对 $l=1,2,\cdots$ 计算下列各步.

2. 计算

$$T_1^{(l)}=\frac{1}{2}\left[T_1^{(l-1)}+\frac{b-a}{2^{l-1}}\sum_{j=1}^{2^{l-1}}f\left(a+(2j-1)\frac{b-a}{2^l}\right)\right].$$

3. 计算表 4.7 的第 $l+1$ 行元素

$$T_{m+1}^{(k-1)}=\frac{4^mT_m^{(k)}-T_m^{(k-1)}}{4^m-1},$$

$$m=1,2,\cdots,l,\quad k=l,l-1,l-2,\cdots.$$

4. 收敛控制

如果表 4.7 的对角线上的最后两相邻元素 $T_m^{(0)}, T_{m-1}^{(0)}$ 满足

$$|T_m^{(0)} - T_{m-1}^{(0)}| < \varepsilon \quad （取绝对误差）$$

或

$$\frac{|T_m^{(0)} - T_{m-1}^{(0)}|}{|T_m^{(0)}|} < \varepsilon \quad （取相对误差），$$

则以 $T_m^{(0)}$ 作为近似值，否则继续到下面一步.

5. l 增加 1 后转到步骤 2 继续做. 为使整个过程不会无限做下去，可设 $l > l^*$ 时计算终止，此时达不到要求，计算不成功.

例 4.4.3 用龙贝格积分法计算

$$I(f) = \int_0^{\pi} \sin x \mathrm{d}x.$$

解 $I(f)$的精确值为 2. 用龙贝格计算过程及计算结果见表 4.8.

表 4.8

①0					
②1.57079633	③2.09439511				
④1.89611890	⑤2.00455976	⑥1.99857073			
⑦1.97423160	⑧2.00026917	⑨1.99998313	⑩2.00000555		
⑪1.99357034	⑫2.00001659	⑬1.99999975	⑭2.00000001	⑮1.99999999	
⑯1.99839336	⑰2.00000103	⑱2.00000000	⑲2.00000000	⑳2.00000000	㉑2.00000000

可以看到，表的第一列是区间逐步分半的复合求积法所得到的结果，虽然计算简单，但收敛很慢，而使用龙贝格积分则得到非常精确的值.

4.5 高斯求积公式

讨论积分

$$I(f) = \int_a^b \rho(x) f(x) \mathrm{d}x \tag{4.5.1}$$

的求积方法，其中积分区间$[a,b]$可以是有限的，也可以是无限

的. $\rho(x)$ 为$[a,b]$上的权函数,即它满足下面三个条件

(1) $\rho(x) \geqslant 0, \forall x \in [a,b]$.

(2) $\int_a^b \rho(x)\mathrm{d}x > 0$.

(3) $\int_a^b |x|^n \rho(x)\mathrm{d}x$ 存在并有限, $\forall n \geqslant 0$.

如果取 $\rho(x) \equiv 1$,那么式(4.5.1)就化为通常的积分.

4.5.1 高斯型求积公式

现在用 n 个不等距的节点 $x_1, x_2, \cdots, x_n$; 其中 $x_k \in [a,b]$ $(k=1,2,\cdots,n)$,对 f 进行插值,则有

$$f(x) = \sum_{k=1}^{n} f(x_k) \frac{\omega_n(x)}{(x-x_k)\omega_n'(x_k)} + f[x,x_1,\cdots,x_n]\omega_n(x), \quad x \in [a,b],$$

其中

$$\omega_n(x) = (x-x_1)(x-x_2)\cdots(x-x_n),$$

$f[x,x_1,\cdots,x_n]$是 n 阶差商.

用权函数 ρ 乘上式并在$[a,b]$上积分得

$$I(f) = \int_a^b \rho(x) f(x)\mathrm{d}x = \sum_{k=1}^{n} A_k f(x_k) + E(f), \tag{4.5.2}$$

其中

$$A_k = \int_a^b \rho(x) \frac{\omega_n(x)}{(x-x_k)\omega_n'(x_k)}\mathrm{d}x, \tag{4.5.3}$$

$$\begin{aligned} E(f) &= I(f) - \sum_{k=1}^{n} A_k f(x_k) \\ &= \int_a^b \rho(x) f[x,x_1,x_2,\cdots,x_n]\omega_n(x)\mathrm{d}x. \end{aligned} \tag{4.5.4}$$

如果 f 取为 $n-1$ 次多项式,则有

$$I(f) = \sum_{k=1}^{n} A_k f(x_k),$$

如果 f 为次数不高于 $2n-1$ 次的多项式，那么 f 的插值多项式的余项中 $f[x,x_1,\cdots,x_n]$ 为次数不高于 $n-1$ 次多项式. 设 $\{g_k\}$ 为 $[a,b]$ 上关于权函数 ρ 的正交多项式族(见 3.4 节)，则有

$$f[x,x_1,\cdots,x_n]=\sum_{k=0}^{n-1}c_kg_k(x),$$

因此

$$\begin{aligned}E(f)&=\int_a^b\rho(x)\omega_n(x)\sum_{k=0}^{n-1}c_kg_k(x)\mathrm{d}x\\&=\sum_{k=0}^{n-1}c_k\int_a^b\rho(x)\omega_n(x)g_k(x)\mathrm{d}x.\end{aligned}$$

如果把非等距节点取成正交多项式 g_n 的根，那么 ω_n 和 g_n 只差一个常数因子，即 $\omega_n=\alpha g_n$，这样，利用正交性得

$$E(f)=\alpha\sum_{k=0}^{n-1}c_k\int_a^b\rho(x)g_n(x)g_k(x)\mathrm{d}x=0.$$

所以，只要把插值节点取作在 $[a,b]$ 上带权 ρ 的正交多项式 g_n 的零点，就可以使具有 n 个节点的求积公式

$$\int_a^b\rho(x)f(x)\mathrm{d}x\approx\sum_{k=1}^{n}A_kf(x_k)\qquad(4.5.5)$$

的代数精度达到 $2n-1$.

定义 4.5.1 若 n 个节点的插值求积公式(4.5.5)的代数精度是 $2n-1$，那么称求积公式是高斯型求积公式. 其节点称为高斯点.

如果节点 x_k 取为正交多项式 g_n 的零点，并设 a_n 为 g_n 中 x^n 的系数，$\beta_n=a_{n+1}/a_n$，$\sigma_n=\int_a^b\rho(x)[g_n(x)]^2\mathrm{d}x$，那么式(4.5.5)对次数小于 $2n$ 的多项式 f 成为等式，系数 A_k 化为

$$A_k=-\frac{\beta_n\sigma_n}{g_n'(x_k)g_{n+1}(x_k)}\quad(k=1,2,\cdots,n).\qquad(4.5.6)$$

定理 4.5.2 对于高斯型求积公式(4.5.5)，若 $f\in C^{2n}[a,b]$，

那么有

$$\int_a^b \rho(x) f(x) \mathrm{d}x - \sum_{k=1}^{n} A_k f(x_k)$$

$$= \frac{1}{(2n)!} f^{(2n)}(\eta) \int_a^b \rho(x) [\omega_n(x)]^2 \mathrm{d}x, \quad \eta \in [a,b],$$

其中 $\omega_n(x) = (x - x_1) \cdots (x - x_n)$.

定理 4.5.3 高斯型求积公式(4.5.5)中的系数 $A_k(k=1, 2, \cdots, n)$全部为正.

令

$$Q_n(f) = \sum_{k=1}^{n} A_k f(x_k),$$

有下面定理.

定理 4.5.4 设$\tilde{f}(x_k)$为 $f(x_k)(k=1,2,\cdots,n)$的近似值,那么有

$$|Q_n(\tilde{f}) - Q_n(f)| \leqslant \max_{1 \leqslant k \leqslant n} |\tilde{f}(x_k) - f(x_k)| \int_a^b \rho(x) \mathrm{d}x.$$

定理 4.5.4 说明采用高斯型求积公式在数值计算中是稳定的. 下面给出收敛性定理.

定理 4.5.5 设 $f \in C[a,b]$,对于高斯型求积公式(4.5.5)有

$$\lim_{n \to \infty} Q_n^{(n)}(f) = \int_a^b \rho(x) f(x) \mathrm{d}x,$$

其中

$$Q_n^{(n)}(f) = \sum_{k=1}^{n} A_k^{(n)} f(x_k^{(n)}),$$

$x_k^{(n)}(k=1,2,\cdots,n)$为高斯点,$A_k^{(n)}$ 为求积系数,上标(n)表示它们依赖于变化着的 n.

由于正交多项式随区间、权函数不同而不同,因此有不同类型的高斯求积公式.

4.5.2 高斯-勒让德求积公式

设区间$[a,b]=[-1,1]$,在$[-1,1]$上取权函数$\rho(x)\equiv 1$,那么相应的正交多项式为勒让德多项式P_n,

$$\mathrm{P}_n(x)=\frac{1}{2^n n!}\frac{\mathrm{d}^n}{\mathrm{d}x^n}[(x^2-1)^n].$$

设$f\in C[-1,1]$,那么高斯求积公式化为

$$\int_{-1}^{1}f(x)\mathrm{d}x=\sum_{k=1}^{n}A_k f(x_k)+R_n[f], \tag{4.5.7}$$

其中高斯点$x_1,x_2,\cdots,x_n$为勒让德多项式$\mathrm{P}_n(x)$的零点,求积公式(4.5.7)称为**高斯-勒让德求积公式**.公式4.5.7中求积系数

$$A_k=\frac{2}{n}\frac{1}{\mathrm{P}_{n-1}(x_k)\mathrm{P}'_n(x_k)}.$$

高斯-勒让德求积公式(4.5.7)中的高斯点x_k,求积系数A_k见4.16节中表4.23.

定理4.5.6 设$f\in C^{2n}[a,b]$,求积公式(4.5.7)的误差

$$R_n[f]=\frac{1}{[(2n)!]^3}\frac{2^{2n+1}}{2n+1}(n!)^4 f^{(2n)}(\eta),\quad \eta\in[-1,1].$$

由表4.23的节点x_k和系数A_k,利用高斯-勒让德求积公式(4.5.7)容易计算出积分的近似值,而且精度相当高.表4.9列出了定理4.5.6的误差,可以看出$f^{(2n)}(\eta)$的系数下降得很快.

表 4.9

n	$E_n(f)$	n	$E_n(f)$
1	$\frac{1}{3}f''(\eta)$	5	$\frac{1}{1237732650}f^{(10)}(\eta)$
2	$\frac{1}{135}f^{(4)}(\eta)$	6	$\frac{1}{648984486150}f^{(12)}(\eta)$
3	$\frac{1}{15750}f^{(6)}(\eta)$	7	$\frac{1}{470050192111500}f^{(14)}(\eta)$
4	$\frac{1}{3472875}f^{(8)}(\eta)$		

例 4.5.7 用牛顿-科茨求积公式和高斯-勒让德求积公式计算积公

$$I(f)=\int_{-1}^{1}\sqrt{x+1.5}\,\mathrm{d}x.$$

解 积分精确值 $I(f)=2.399529$. 计算结果见表 4.10.

表 4.10

n	高斯-勒让德	Newton-Cotes
2	2.401843	2.288246
3	2.399709	2.395742

由表 4.10 可以看出,在节点数目相等的情况下,高斯-勒让德求积公式的结果更为精确.

对于权函数 $\rho=1$ 的在任意有限区间$[a,b]$上的积分,可以通过自变数的线性变换

$$t=\frac{b-a}{2}x+\frac{a+b}{2}$$

把积分化为标准区间$[-1,1]$上的积分

$$\int_a^b f(x)\mathrm{d}x=\frac{b-a}{2}\int_{-1}^{1}f\left(\frac{a+b+(b-a)x}{2}\right)\mathrm{d}x.$$

例 4.5.8 用高斯-勒让德求积公式计算积分

$$I(f)=\int_0^{\pi}\mathrm{e}^x\cos x\,\mathrm{d}x.$$

解 积分精确值为 -12.0703463164. 用高斯-勒让德公式计算的结果见表 4.11,可以看出这是相当好的(这个积分在例 4.3.6 中曾分别用复合梯形公式和复合辛普森公式计算过).

表 4.11

n	$I_n(f)$	$E_n(f)$
2	−12.33621046570	2.66×10^{-1}
3	−12.12742045017	5.71×10^{-2}
4	−12.07018949029	-1.57×10^{-4}
5	−12.07032853589	-1.78×10^{-5}
6	−12.07034633110	1.47×10^{-8}
7	−12.07034631753	1.14×10^{-9}
8	−12.07034631639	$<5.0\times10^{-12}$

高斯-勒让德求积公式的误差由定理 4.5.6 给出，但是在很多应用中，用被积函数求导的办法来估计误差是不方便的. 此外有的被积函数没有高阶导数或不可导，因而不能采用这样的方法来估计误差. 下面两种方法在估计求积公式的误差是经常采用的.

(1) 用更高阶的高斯-勒让德求积公式来检验其结果.

(2) 把积公区间分成几个子区间，在这些子区间上采用同样的高斯-勒让德求积公式.

例 4.5.9 用方法(1)来计算积分 $\int_1^2 \frac{1}{x}\mathrm{d}x$.

解 以 I_n 表示用 n 个节点的高斯-勒让德求积公式计算的结果，有

$$I_3 = 0.693122,$$
$$I_4 = 0.693146,$$
$$I_5 = 0.693147.$$

可以看出，I_4 和 I_5 相当靠近，因此取 I_5 作为积分的近似值.

应该注意，数值接近有些是虚假的，因此还必须用更高阶公式来检验. 例如计算积分

$$I = \int_{-1}^{1} \frac{1}{x^4 + x^2 + 0.9}\mathrm{d}x,$$

得

$$I_3 = 1.585026,$$
$$I_4 = 1.585060,$$

由此似乎可以看出，取1.585 作为近似值就有 4 位有效数字，但用五个节点的高斯-勒让德求积公式就看出上面结果不对. 积分 I 的精确值应是 1.582233.

方法(2)的使用可以取不同的方式. 最简单的就是先把积分区间分成两个相等的子区间，而在每个子区间上采用同样的高斯-勒让德求积分式.

例 4.5.10 用方法(2) 来计算积分 $I=\int_1^2 \frac{1}{x}\mathrm{d}x$.

解 先把积分区间二等分，有

$$\begin{aligned} I &= \int_1^{1.5} \frac{1}{x}\mathrm{d}x + \int_{1.5}^2 \frac{1}{x}\mathrm{d}x \\ &= \int_{-1}^1 \frac{1}{5+u}\mathrm{d}u + \int_{-1}^1 \frac{1}{7+u}\mathrm{d}u. \end{aligned}$$

对上面的每个积分应用三点高斯-勒让德求积公式有

$$I = 0.405464 + 0.287682 = 0.693146.$$

分区间的过程可以继续下去，直到子区间(a_k,b_k)上的积分近似值与$\left(a_k,\frac{1}{2}(a_k+b_k)\right)$及$\left(\frac{1}{2}(a_k+b_k),b_k\right)$上的积分近似值之和的差在允许的误差范围之内.

上述计算过程的缺点是前次计算的函数值没有利用，为克服这个缺点可采用鲁宾逊(Robinson)方法. 其方法如下：先用三点高斯-勒让德求积公式进行计算，然后把积分区间分为三个子区间，使其每个区间的中点为求积的节点 $x_k(k=1,2,3)$，于是区间$(-1,1)$分为$(-1,-\alpha)$，$(-\alpha,\alpha)$，$(\alpha,1)$，其中 $\alpha=2\sqrt{0.6}-1$. 最后把三点高斯-勒让德求积公式应用到这三个区间.



例 4.5.11 用 Robinson 技巧计算 $I=\int_1^2 \frac{1}{x}\mathrm{d}x$.

解

$$
\begin{aligned}
I &= \int_{-1}^{1} \frac{1}{3+u}\mathrm{d}u \\
&= \int_{-1}^{-\alpha} \frac{1}{3+u}\mathrm{d}u + \int_{-\alpha}^{\alpha} \frac{1}{3+u}\mathrm{d}u + \int_{\alpha}^{1} \frac{1}{3+u}\mathrm{d}u \\
&= \beta\int_{-1}^{1} \frac{1}{3-\gamma+\beta v}\mathrm{d}v + \alpha\int_{-1}^{1} \frac{1}{3+\alpha v}\mathrm{d}v + \\
&\quad \beta\int_{-1}^{1} \frac{1}{3+\gamma+\beta v}\mathrm{d}v,
\end{aligned}
$$

其中 $\alpha=2\sqrt{0.6}-1,\gamma=\sqrt{0.6},\beta=1-\gamma$.

计算(用三个节点的高斯-勒让德求积公式)得

$$
\begin{aligned}
I &\approx 0.203270 + 0.370303 + 0.119574 \\
&= 0.693147.
\end{aligned}
$$

这样的过程可以继续下去,如果在子区间(a_k,b_k)上直接用三点高斯-勒让德求积公式,所得到的积分值与在该区间上用鲁宾逊方法的积分值之差在允许误差范围内,即得计算结果.

4.5.3 高斯-切比雪夫求积公式

以区间$[a,b]=[-1,1]$,权函数 $\rho(x)=\frac{1}{\sqrt{1-x^2}}$的正交多项式序列为切比雪夫多项式

$$T_n(x) = \cos(n \arccos x)$$

构成的序列$\{T_n\}$. 设 $f\in C[-1,1]$,高斯求积公式为

$$\int_{-1}^{1} \frac{1}{\sqrt{1-x^2}}f(x)\mathrm{d}x = \sum_{k=1}^{n} A_k f(x_k) + R_n[f], \qquad (4.5.8)$$

其中 x_k 为 n 次切比雪夫多项式 $T_n(x)$的零点,求积公式(4.5.8)称为高斯-切比雪夫求积公式.

$$x_k = \cos\frac{(2k-1)\pi}{2n} \quad (k=1,2,\cdots,n).$$

$$A_k = \frac{\pi}{n}.$$

定理 4.5.12 设 $f\in C^{2n}[-1,1]$，高斯-切比雪夫求积公式(4.5.8)的误差为

$$R_n[f] = \frac{2\pi}{2^{2n}(2n)!}f^{(2n)}(\eta), \quad \eta\in[-1,1].$$

例 4.5.13 计算积分

$$I(f) = \int_{-1}^{1}\frac{e^x}{\sqrt{1-x^2}}dx,$$

使其精确到 10^{-6}.

解 利用定理 4.5.12，$f(x)=e^x$，$f^{2n}(\eta)=e^\eta$，

$$R_n[f] = \frac{2\pi}{2^{2n}(2n)!}e^\eta, \quad \eta\in[-1,1],$$

从而有估计

$$|R_n(f)| \leqslant \frac{2\pi}{2^{2n}(2n)!}e \equiv B_n,$$

对于 $n=4$，$B_4=1.66\times10^{-6}$，而对于 $n=5$，$B_5=4.6\times10^{-9}$，因此取 $n=5$. 利用求积公式得

$$I(f)\approx 3.977463.$$

4.5.4 高斯-拉盖尔求积公式

区间 $[a,b]=[0,+\infty)$，权函数 $\rho(x)=e^{-x}$，$x\in[0,+\infty)$ 此时高斯求积公式为

$$\int_0^\infty e^{-x}f(x)dx = \sum_{k=1}^{n}A_kf(x_k)+R_n[f], \qquad (4.5.9)$$

其中节点 $x_k(k=1,2,\cdots,n)$ 是 n 次拉盖尔多项式

$$L_n(x) = e^x\frac{d^n}{dx^n}(x^ne^{-x}) \quad (n=0,1,\cdots)$$

的零点，而系数

$$A_k = \frac{(n!)^2}{L'_n(x_k)L_{n+1}(x_k)}.$$

公式(4.5.9)称为**高斯-拉盖尔求积公式**. 节点 x_k 与系数 A_k 可见4.16节中表4.24.

定理 4.5.14 设 $f \in C^{2n}(0,\infty)$，高斯-拉盖尔求积公式的误差为

$$R_n[f] = \frac{(n!)^2}{(2n)!} f^{(2n)}(\eta), \quad \eta \in [0,\infty).$$

例 4.5.15 用高斯-拉盖尔求积公式(4.5.9)计算积分

$$I(f) = \int_0^{\infty} e^{-x}\sin x dx.$$

解 解析求解 $I(f)=0.5$. 用高斯-拉盖尔求积公式 $n=2$，$I_2(f)=0.43246$. $n=3$，$I_3(f)=0.49603$.

4.5.5 高斯-埃尔米特求积公式

区间$[a,b]=(-\infty,\infty)$，权函数 $\rho(x)=e^{-x^2}$，$x\in(-\infty,\infty)$. 此时高斯求积公式

$$\int_{-\infty}^{\infty} e^{-x^2} f(x)dx = \sum_{k=1}^{n} A_k f(x_k) + R_n[f], \quad (4.5.10)$$

其中节点 $x_k(k=1,2,\cdots,n)$ 为 n 次埃尔米特(Hermite)多项式

$$H_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n} e^{-x^2}$$

的零点，而系数

$$A_k = \frac{-2^{n+1} n! \sqrt{\pi}}{H'_n(x_k) H_{n+1}(x_k)}.$$

求积公式(4.5.10)称为**高斯-埃尔米特(Gauss-Hermite)求积公式**. 节点 x_k 及系数 A_k 可见4.16节中表4.23.

定理 4.5.16 设 $f\in C^{2n}(-\infty,\infty)$，那么高斯-埃尔米特求积公式的误差为

$$R_n[f]=\frac{n!\sqrt{\pi}}{2^n(2n)!}f^{(2n)}(\eta),\quad \eta\in(-\infty,\infty).$$

4.6 预先给定节点的高斯求积公式

在 n 个节点的高斯求积公式中，节点是取为带权函数 $\rho(x)$ 的 n 次正交多项式的零点. 但是在有些应用中，希望一个端点或两个端点预先固定，由此就要对高斯求积公式作一定的修正.

最常用的情况是求积区间为 $[-1,1]$. 把区间 $[-1,1]$ 的一个端点 -1 或 1 预先固定的高斯求积公式称为**高斯-拉道**(Gauss-Radau)**求积公式**. 把区间 $[-1,1]$ 的两个端点 -1 和 1 都预先固定的高斯求积公式称为**高斯-洛巴托**(Gauss-Lobatto)**求积公式**.

4.6.1 高斯-拉道求积公式

积分 $I(f)=\int_{-1}^{1}f(x)\mathrm{d}x$ 的高斯-拉道求积公式为

$$I(f)=\frac{2}{n^2}f(-1)+\sum_{k=2}^{n}A_kf(x_k)+R_n[f],\quad (4.6.1)$$

其中 $x_k(k=2,3,\cdots,n)$ 是多项式

$$\phi_{n-1}(x)=\frac{1}{x+1}[\mathrm{P}_{n-1}(x)+\mathrm{P}_n(x)],\quad x\in[-1,1]$$

的零点，$\mathrm{P}_n(x)$ 为 n 次勒让德多项式，

$$A_k=\frac{1}{1-x_k}\frac{1}{[\mathrm{P}'_{n-1}(x_k)]^2}.$$

求积公式(4.6.1)中仅有一个端点 $x=-1$ 预先固定. 求积公式对于 $n=2,3,4,5$ 的节点和系数见表 4.12.

表 4.12

n	$x_k(1\leqslant k\leqslant n)$	$A_k(1\leqslant k\leqslant n)$
2	−1	0.5
	0.333333	1.5
3	−1	0.222222
	−0.289898	0.752806
	0.689898	1.024972
4	−1	0.125000
	−0.575319	0.657689
	0.181066	0.776387
	0.822824	0.440925
5	−1	0.080000
	−0.720480	0.446207
	−0.167181	0.623653
	0.446314	0.562712
	0.885792	0.287427

定理 4.6.1 设 $f\in C^{2n-1}[-1,1]$，那么高斯-拉道求积公式的误差为

$$R_n[f]=\frac{2^{2n-1}n}{[(2n-1)!]^3}[(n-1)!]^4 f^{(2n-1)}(\eta),\quad \eta\in[-1,1].$$

4.6.2 高斯-洛巴托求积公式

设 $x_1=-1,x_n=1$ 预先固定. 积分 $I(f)=\int_{-1}^{1}f(x)\mathrm{d}x$ 的高斯-洛巴托求积公式为

$$\int_{-1}^{1}f(x)\mathrm{d}x=\frac{2}{n(n-1)}[f(-1)+f(1)]+\sum_{k=2}^{n-1}A_k f(x_k)+R_n[f],\tag{4.6.2}$$

其中 $x_k(k=2,3,\cdots,n-1)$ 是多项式 $\mathrm{P}'_{n-1}(x)$ 的零点，$\mathrm{P}_{n-1}(x)$ 为

$n-1$次勒让德多项式.

$$A_k=\frac{2}{n(n-1)[\mathrm{P}_{n-1}(x_k)]^2}\quad (x_k\neq\pm 1).$$

高斯-洛巴托求积公式的节点 x_k 和系数 A_k 见表 4.13.

表 4.13

n	x_k	A_k
3	0	$\frac{4}{3}$
	± 1	$\frac{1}{3}$
4	± 0.447214	$\frac{5}{6}$
	± 1	$\frac{1}{6}$
5	0	$\frac{32}{45}$
	± 0.654654	$\frac{49}{90}$
	± 1	$\frac{1}{10}$
6	± 0.285232	0.554858
	± 0.765055	0.378475
	± 1	0.066667

定理 4.6.2 设 $f\in C^{2n-2}[-1,1]$,那么高斯-洛巴托求积公式(4.6.3)的误差为

$$R_n[f]=-\frac{n(n-1)^3 2^{2n-1}[(n-2)!]^4}{(2n-1)[(2n-2)!]^3}f^{(2n-2)}(\eta),\quad \eta\in[-1,1]$$

在高斯-洛巴托求积公式中,被积函数 $f(x)$ 要在 $x=\pm 1$ 处计算函数值,因此失去了两个自由度(注意,高斯-洛巴托求积公式中要在 $x=-1$ 或 $x=1$ 处计算 $f(x)$ 的值,因而失去了一个自由度).由此可知,n 个节点的高斯-洛巴托求积公式的代数精度为

$2n-3$. 一般 n 个节点的高斯-勒让德求积公式的代数精度为 $2n-1$. 然而，如果知道被积函数 $f(x)$ 在区间的端点上取值为 0，则宜使用高斯-洛巴托求积公式（对高斯-拉道求积公式也类似）. 这种情况下，$n+2$ 个节点的高斯-洛巴托求积公式只需计算 n 个函数 $f(x)$ 的值，而且具有 $2n+1$ 次代数精度. 注意，n 个节点的高斯-勒让德求积公式也需要计算 n 个函数 $f(x)$ 的值，但代数精度仅为 $2n-1$.

例 4.6.3 用 5 个节点的高斯-洛巴托求积公式计算积分

$$I(f)=\int_{-1}^{1}\cos\frac{\pi x}{2}\mathrm{d}x.$$

解 $I(f)$ 的精确值为 $\frac{4}{\pi}$，取 5 位小数为 1.27324. 由于被积函数在积分区间 $[-1,1]$ 的端点取值为 0，因此 5 个节点的高斯-洛巴托求积公式仅需要计算 3 个函数值.

$$I(f)\approx\frac{49}{90}\cos\frac{1}{2}\pi\times(-0.654654)+\frac{32}{45}\cos 0+\frac{49}{90}\cos\frac{\pi}{2}\times(0.654654)=1.2732.$$

利用高斯-勒让德 3 个节点的求积公式，同样要计算 3 个函数值，其结果是

$$I(f)\approx 1.27412.$$

两者相比，显然高斯-洛巴托求积公式更为精确.

4.7 切比雪夫求积法

对于积分

$$I(f)=\int_{-1}^{1}f(x)\mathrm{d}x,$$

如果 f 是 $[-1,1]$ 上的有界变差的连续函数，那么 f 可以展成一

致收敛的切比雪夫多项式的级数

$$f(x)=\frac{1}{2}A_0T_0(x)+A_1T_1(x)+A_2T_2(x)+\cdots \quad (4.7.1)$$

其中 $T_k(x)(k=0,1,\cdots)$是多项式，

$$A_k=\frac{2}{\pi}\int_{-1}^{1}\frac{f(x)T_k(x)}{\sqrt{1-x^2}}\mathrm{d}x=\frac{2}{\pi}\int_0^{\pi}f(\cos\theta)\cos k\theta\mathrm{d}\theta.$$

$$k=0,1,2,\cdots \quad (4.7.2)$$

切比雪夫求积法就是先将被积函数 $f(x)$展成式(4.7.1)，然后取级数的部分和

$$S_n(x)=\frac{1}{2}A_0+A_1T_1(x)+\cdots+A_nT_n(x),$$

$$x\in[-1,1] \quad (4.7.3)$$

来近似 $f(x)$.

直接用公式(4.7.2)计算系数，一般相当困难，因此通常采用近似式 $S_n(x)$. 下面给出两种方法.

(1) $S_n(x)$取为

$$\hat{S}_n(x)=\sum_{r=0}^{n}{}'b_rT_r(x), \quad (4.7.4)$$

其中求和号 $\sum$ 右上方的"′"表示第一项要用$\frac{1}{2}$来乘. 而

$$b_r=\frac{2}{n+1}\sum_{k=0}^{n}f(x_k)T_r(x_k). \quad (4.7.5)$$

上式中 x_k 取为

$$x_k=\cos\left(\frac{2k+1}{n+1}\cdot\frac{\pi}{2}\right) \quad (k=0,1,\cdots,n). \quad (4.7.6)$$

利用式(4.7.4)，有

$$I(f)=\int_{-1}^{1}f(x)\mathrm{d}x\approx\int_{-1}^{1}\hat{S}_n(x)\mathrm{d}x=\sum_{r=0}^{n}{}'\int_{-1}^{1}b_rT_r(x)\mathrm{d}x.$$

由于

$$\int T_r(x)\mathrm{d}x = \begin{cases} T_1(x) + c_0, & r = 0, \\ \dfrac{1}{4}T_2(x) + c_1, & r = 1, \\ \dfrac{1}{2}\left[\dfrac{T_{r+1}(x)}{r+1} - \dfrac{T_{r-1}(x)}{r-1}\right] + c_2, & r > 1, \end{cases}$$

其中积分常数 c_0, c_1 和 c_2 已作了调整.

因此得

$$\int_{-1}^{1} T_r(x)\mathrm{d}x = \begin{cases} 0, & r\text{ 为奇数}, \\ \dfrac{1}{r+1} - \dfrac{1}{r-1}, & r\text{ 为偶数}. \end{cases}$$

把上式代入 $I(f)$ 的近似式有

$$I(f) = \int_{-1}^{1} f(x)\mathrm{d}x \approx \sum_{r=0}^{\left[\frac{1}{2}n\right]} B_{2r+1}, \tag{4.7.7}$$

其中$\left[\dfrac{1}{2}n\right]$表示小于等于$\dfrac{1}{2}n$的最大整数.

$$B_{2s+1} = \frac{b_{2s}}{2s+1}, \quad s = \left[\frac{1}{2}n\right],$$

$$B_{2r+1} = \frac{b_{2r} - b_{2r+2}}{2r+1} \quad (r = 0, 1, \cdots, s-1).$$

(2) S_n 的另一种近似式取为

$$\overline{S}_n(x) = \sum_{r=0}^{n}{}'' c_r T_r(x) \tag{4.7.8}$$

$$= \frac{1}{2}c_0 T_0(x) + c_1 T_1(x) + \cdots + c_{n-1}T_{n-1}(x) + \frac{1}{2}c_n T_n(x)$$

其中求和号 $\sum$ 右上方的"$''$"表示在和式中的第一项和最后一项用$\dfrac{1}{2}$来乘.

$$c_r = \frac{2}{n}\sum_{k=0}^{n}{}'' f(x_k) T_r(x_k), \tag{4.7.9}$$

式中“″”意义同前，$x_k=\cos\dfrac{k\pi}{n}(k=0,1,\cdots,n)$.

求积公式为

$$I(f)=\int_{-1}^{1}f(x)\mathrm{d}x\approx\int_{-1}^{1}\overline{S}_n(x)\mathrm{d}x=\sum_{r=0}^{\left[\frac{1}{2}n\right]}c_{2r+1},\quad(4.7.10)$$

其中

$$c_{2s+1}=\frac{1}{2s+1}ac_{2s},\quad c_{2s-1}=\frac{1}{2s-1}(c_{2s-2}-ac_{2s}),$$

$$s=\left[\frac{1}{2}n\right],\quad a=\begin{cases}\dfrac{1}{2}, & 2s=n,\quad n\geqslant 2,\\ 1, & 2s\neq n,\end{cases}$$

$$c_{2r+1}=\frac{1}{2r+1}(c_{2r}-c_{2r+2})\quad(r=0,1,\cdots,s-2).$$

例 4.7.1 用切比雪夫求积法计算积分

$$I(f)=\int_{1}^{2}\frac{1}{x}\mathrm{d}x.$$

解 $I(f)=\ln 2\approx 0.69315$.

先用线性变换 $x=\dfrac{1}{2}(u+3)$ 把积分区间$[1,2]$变为$[-1,1]$，这样有

$$\int_{1}^{2}\frac{1}{x}\mathrm{d}x=\int_{-1}^{1}\frac{1}{u+3}\mathrm{d}u.$$

采用方法(2)，取 $n=2$，则有

$$c_0=\frac{1}{2}f(x_0)T_0(x_0)+f(x_1)T_0(x_1)+\frac{1}{2}f(x_2)T_0(x_2),$$

其中 $x_0=-x_2=1,x_1=0$. 计算得 $c_0=\dfrac{17}{24}$. 同理有 $c_2=\dfrac{1}{24}$. 于是 $c_3=\dfrac{1}{144}$，$c_1=\dfrac{11}{16}$，由此得 I 的近似值

$$I\approx\left(\frac{11}{16}+\frac{1}{144}\right)=\frac{25}{36}=0.69444.$$

取 $n=4$，有

$$c_0 = 0.70710, \quad c_2 = 0.00674, \quad c_4 = 0.00122,$$

因此得

$$c_5 = 0.0012, \quad c_3 = 0.00674, \quad c_1 = 0.68627.$$

这样可以得出 I 的近似值

$$I \approx c_1 + c_3 + c_5 = 0.69313.$$

这是一个很好的近似结果.

4.8 三次样条函数求积法

用代数插值多项式 P_n 来代替被积函数 f，可以得到许多实用的求积公式.用三次样条函数 S 来代替被积函数 f，以求得到新的求积公式，这种方法称为**三次样条函数求积**（numerical integration by cubic spline).

为计算定积分

$$I(f) = \int_a^b f(x)\mathrm{d}x,$$

可先将区间$[a,b]$分成 n 等分，其节点为

$$x_k = a + kh \quad (k = 0,1,\cdots,n),$$

其中 $h=\frac{b-a}{n}$，而且在节点的两端处各延拓一点，$x_{-1}=a-h$，$x_{n+1}=a+(n+1)h$.

设 S 为被积函数 f 的三次样条插值逼近，那么对任 $x\in[a,b]$有

$$f(x) \approx S(x) \approx \sum_{k=-1}^{n+1} c_k \Omega_3\left(\frac{x-x_k}{h}\right).$$

由此可以得到

$$I(f) \approx \int_a^b S(x)\mathrm{d}x = \sum_{k=-1}^{n+1} c_k \int_a^b \Omega_3\left(\frac{x-x_k}{h}\right)\mathrm{d}x. \quad (4.8.1)$$

对于 $n\geqslant 3$，有

$$
\begin{aligned}
I(f) \approx \int_a^b S(x)\mathrm{d}x = I_n(f) \\
&= \frac{h}{24}(c_{-1} + c_{n+1}) + \frac{h}{2}(c_0 + c_n) + \\
&\quad \frac{23}{24}h(c_1 + c_{n-1}) + h\sum_{k=2}^{n-2} c_k. \qquad (4.8.2)
\end{aligned}
$$

4.8.1 一般情况的求积公式

为了确定公式(4.8.2)中的 $c_k(k=-1,0,\cdots,n+1)$，可用三次样条函数的第一边界条件来解决. 由三次样条函数的第一边界条件的条件可以得到如下关系：

$$
\begin{cases}
f(x_0) = \sum_{k=-1}^{n+1} c_k \Omega_3(-k) \\
\qquad = c_0\Omega_3(0) + c_1\Omega_3(-1) + c_{-1}\Omega_3(1) \\
\qquad = \frac{2}{3}c_0 + \frac{1}{6}c_1 + \frac{1}{6}c_{-1}, \\
f(x_n) = \sum_{k=-1}^{n+1} c_k \Omega_3(n-k) \\
\qquad = c_n\Omega_3(0) + c_{n+1}\Omega_3(-1) + c_{n-1}\Omega_3(1) \\
\qquad = \frac{2}{3}c_n + \frac{1}{6}c_{n+1} + \frac{1}{6}c_{n-1}, \\
f'(x_0) = \frac{1}{h}\sum_{k=-1}^{n+1} c_k \Omega'_3(-k) = \frac{c_1 - c_{-1}}{2h}, \\
f'(x_n) = \frac{1}{h}\sum_{k=-1}^{n+1} c_k \Omega'_3(n-k) = \frac{c_{n+1} - c_{n-1}}{2h}, \\
f(x_k) = \sum_{j=-1}^{n+1} c_j \Omega_3(k-j) \\
\qquad = c_k\Omega_3(0) + c_{k+1}\Omega_3(-1) + c_{k-1}\Omega_3(1) \\
\qquad = \frac{2}{3}c_k + \frac{1}{6}c_{k+1} + \frac{1}{6}c_{k-1} \quad (k = 1,2,\cdots,n-1).
\end{cases} \qquad (4.8.3)
$$

由公式(4.8.3)的各式可以得到

$$h\sum_{k=1}^{n-1}f(x_k)=h\left[\sum_{k=2}^{n-2}c_k\right]+\frac{h}{6}(c_0+c_n)+$$

$$\frac{5}{6}h(c_1+c_{n-1})\ \frac{h}{2}[f(x_0)+f(x_n)]$$

$$=\frac{h}{3}(c_0+c_n)+\frac{h}{12}(c_1+c_{n-1})+\frac{h}{12}(c_{-1}+c_{n+1}).$$

由上面两式得

$$\frac{1}{2}h\Big[f(x_0)+f(x_n)+2\sum_{k=1}^{n-1}f(x_k)\Big]$$

$$=I_n(f)+\frac{h}{24}(c_{-1}-c_1+c_{n+1}-c_{n-1}),$$

其中 $I_n(f)$由式(4.8.2)所定义.再利用公式(4.8.3)的第三式和第四式可得到求积公式

$$I(f)\approx I_n(f)$$

$$=\frac{h}{2}\Big(f(x_0)+f(x_n)+2\sum_{k=1}^{n-1}f(x_k)\Big)+$$

$$\frac{h^2}{12}\big(f'(x_0)-f'(x_n)\big). \tag{4.8.4}$$

这就是三次样条函数第一边界条件的求积公式.

定理 4.8.1 设 $f\in C^4[a,b]$,那么求积公式(4.8.4)的误差为

$$R_n[f]=I(f)-I_n(f)=\frac{b-a}{720}h^4f^{(4)}(\eta),\quad \eta\in[a,b].$$

4.8.2 简单情况的求积公式

公式(4.8.2)仅对 $n\geqslant 3$ 时才成立,因此对于 $n=1,n=2$ 这两种特殊情况必须单独讨论.当 $n=1$ 时,

$$\int_a^b\Omega_3\Big(\frac{x-x_0}{h}\Big)\mathrm{d}x=\int_a^b\Omega_3\Big(\frac{x-x_1}{h}\Big)\mathrm{d}x=h\int_0^1\Omega_3(x)\mathrm{d}x=\frac{11}{24}h,$$

$$\int_a^b \Omega_3\left(\frac{x-x_{-1}}{h}\right)\mathrm{d}x = \int_a^b \Omega_3\left(\frac{x-x_2}{h}\right)\mathrm{d}x = h\int^2 \Omega_3(x)\mathrm{d}x = \frac{1}{24}h.$$

由式(4.8.1)得

$$I(f) \approx \frac{h}{24}(c_{-1}+c_2) + \frac{11}{24}h(c_0+c_1).$$

对于三次样条第一边界条件，$c_k(k=-1,0,1,2)$可由以下方程组确定

$$\begin{cases} -c_{-1}+c_1 = 2hf'(x_0), \\ c_{-1}+4c_0+c_1 = 6f(x_0), \\ c_0+4c_1+c_2 = 6f(x_1), \\ -c_0+c_2 = 2hf'(x_1). \end{cases}$$

解之，得

$$\begin{cases} c_{-1} = (2f(x_1)-f(x_0)) - \dfrac{h}{3}(2f'(x_1)+7f'(x_0)), \\ c_0 = (2f(x_0)-f(x_1)) + \dfrac{h}{3}(2f'(x_0)+f'(x_1)), \\ c_1 = (2f(x_1)-f(x_0)) - \dfrac{h}{3}(2f'(x_1)+f'(x_0)), \\ c_2 = (2f(x_0)-f(x_1)) + \dfrac{h}{3}(2f'(x_0)+7f'(x_1)). \end{cases}$$

由此得到

$$\int_a^b f(x)\mathrm{d}x \approx \frac{h}{2}(f(x_0)+f(x_1)) + \frac{h^2}{12}(f'(x_0)-f'(x_1)). \tag{4.8.5}$$

与 $n=1$ 的推导相似，可以得到 $n=2$ 的求积公式

$$\int_a^b f(x)\mathrm{d}x \approx \frac{1}{2}h(f(x_0)+2f(x_1)+f(x_2)) + \frac{1}{12}h^2(f'(x_0)-f'(x_2)). \tag{4.8.6}$$

例 4.8.2 用三次样条函数求积法计算积分

$$\int_{0.5}^{1} \sqrt{x}\mathrm{d}x.$$

解 $I(f)=\frac{2}{3}\left(1-\left(\frac{1}{\sqrt{2}}\right)^3\right)\approx 0.4309644.$

对于 $n=1$,利用公式(4.8.5)有

$$\begin{aligned}\int_{0.5}^{1} \sqrt{x}\mathrm{d}x &\approx \frac{0.5}{2}\left[\sqrt{0.5}+1\right]+\frac{0.5^2}{12}\left[\frac{1}{2\sqrt{0.5}}-\frac{1}{2}\right]\\ &= 0.43109142.\end{aligned}$$

对于 $n=2$,利用公式(4.8.6)有

$$\int_{0.5}^{1} \sqrt{x}\mathrm{d}x \approx 0.43097338.$$

对于 $n=3$,利用公式(4.8.4)有

$$\int_{0.5}^{1} \sqrt{x}\mathrm{d}x \approx 0.43096623.$$

可以看出,三次样条求积公式对于 $n=1,2,3$ 分别有三位、四位、五位有效数字.

4.9 自适应积分法

被积函数在整个积分区间$[a,b]$上变化不是很均匀的,如在某点附近函数变化非常急剧,而在其余地方的变化比较平缓.这种情况用等距剖分小区间的复合求积公式不很适合.为了使计算达到预定的精度又要节省工作量,则可以在函数变化急剧的部分增多节点,即子区间分得细,而在函数变化平缓的地方减少节点,即子区间分得大.这个方法称为**自适应积分法**(adaptive quadrature methods).

4.9.1 自适应辛普森方法

采用逐次将区间二等分的方法，为写法统一，将积分区间$[a,b]$记为$[a,a+h]$，其中$h=b-a$为区间长度，称原区间为0级区间. 在区间$[a,h]$上采用辛普森公式(4.2.9)，把结果记作

$$S_{a,a+h}^{(1)}=\frac{h}{6}\left[f(a)+4f\left(a+\frac{h}{2}\right)+f(a+h)\right]. \tag{4.9.1}$$

将区间分成两个相等的子区间$\left[a,a+\frac{h}{2}\right]$和$\left[a+\frac{h}{2},a+h\right]$，这两个子区间称为1级子区间，其长度为$\frac{h}{2}$. 在每个1级子区间上采用辛普森公式计算积分，然后相加并令

$$S_{a,a+h}^{(2)}=S_{a,a+\frac{h}{2}}^{(1)}+S_{a+\frac{h}{2},a+h}^{(1)}, \tag{4.9.2}$$

再将1级子区间中的一个或所有的两个二等分，所得的子区间称为2级子区间，其长度为$\frac{1}{2^2}h,\cdots$. 如此继续下去，最后将区间$[a,a+h]$分成n个子区间$[a_i,a_{i+1}]$$(i=0,1,\cdots,n-1)$，这样有

$$\begin{aligned} a=a_0<a_1<\cdots<a_i<a_{i+1}<\cdots<a_n \\ =b=a+h. \end{aligned} \tag{4.9.3}$$

子区间的长度一般是不同的，如果子区间$[a_i,a_{i+1}]$是r级，则其长度为

$$a_{i+1}-a_i=\frac{h}{2^r}.$$

实际上，区间的划分(4.9.3)是根据函数的变化情况而定的，函数变化平缓的地方，子区间大，函数变化急剧的地方，子区间就小.

设$S_{a,a+h}$表示$I(f)=\int_a^b f(x)\mathrm{d}x$的近似值，那么有

$$S_{a,a+h}=\sum_{i=0}^{n-1}S_{a_i,a_{i+1}}^{(2)}.$$

若计算 $I(f)$ 的允许误差为 ϵ，则需有

$$|S_{a,a+h}-I(f)|<\epsilon.$$

如果记 $I_{\alpha,\beta}(f)=\int_{\alpha}^{\beta}f(x)\mathrm{d}x$，则上式化为

$$\left|\sum_{i=0}^{n-1}(S_{a_i,a_{i+1}}^{(2)}-I_{a_i,a_{i+1}}(f))\right|<\epsilon. \tag{4.9.4}$$

如果取每个 r 级子区间上误差控制为 $\frac{1}{2^r}\epsilon$，即取 $[a_i,a_{i+1}]$ 为 r 级子区间，则当

$$|S_{a_i,a_{i+1}}^{(2)}-I_{a_i,a_{i+1}}(f)|<\frac{1}{2^r}\epsilon \tag{4.9.5}$$

时，式(4.9.4)可以得到满足.

要直接验证式(4.9.5)是否成立是相当困难的，可以采用间接方法. 设 f 在 $[a,a_{i+1}]$ 上为五次连续可微，则辛普森求积公式的误差

$$\begin{aligned}I_{a_i,a_{i+1}}(f)-S_{a_i,a_{i+1}}^{(1)}&=-\frac{(a_{i+1}-a_i)^5}{90}f^{(4)}(\eta_1)\\&=-\frac{1}{90}(a_{i+1}-a_i)^5f^{(4)}\left(a_i+\frac{a_{i+1}-a_i}{2}\right)+\\&\quad o((a_{i+1}-a_i)^5)\qquad(a_i<\eta_1<a_{i+1}),\end{aligned}$$

$$\begin{aligned}I_{a_i,a_{i+1}}(f)-S_{a_i,a_{i+1}}^{(2)}&=-\frac{1}{1440}(a_{i+1}-a_i)^5f^{(4)}(\eta_2)\\&=-\frac{1}{1440}(a_{i+1}-a_i)^5f^{(4)}\left(a_i+\frac{a_{i+1}-a_i}{2}\right)+\\&\quad o((a_{i+1}-a_i)^5)\qquad(a_i<\eta_2<a_{i+1}).\end{aligned}$$

如果忽略 $o((a_{i+1}-a_i)^5)$，则有

$$\left|I_{a_i,a_{i+1}}(f)-S_{a_i,a_{i+1}}^{(2)}\right|=\frac{1}{15}\left|S_{a_i,a_{i+1}}^{(2)}-S_{a_i,a_{i+1}}^{(1)}\right|,$$

因此，如果

$$\left|S_{a_i,a_{i+1}}^{(2)}-S_{a_i,a_{i+1}}^{(1)}\right|<15\epsilon\frac{1}{2^r}, \tag{4.9.6}$$

那么式(4.9.5)就满足,从而式(4.9.4)也满足.

在逐次二等分区间的过程中,可以根据不等式(4.9.6)判断是否要将一个 r 级子区间继续分成两个相等的 $r+1$ 级子区间.如果式(4.9.6)成立,则认为在子区间$[a_i,a_{i+1}]$上已达到计算的精确度,因而可以再考虑与$[a_i,a_{i+1}]$右相邻的那个子区间;否则,将继续分$[a_i,a_{i+1}]$为两个相等的 $r+1$ 级子区间.

在实际计算中,式(4.9.6)的右边经常用 $10\varepsilon\frac{1}{2^r}$来代替.

4.9.2 计算步骤

将区间$[a,a+h]$,$h=b-a$,分成两个相等的 1 级子区间$\left[a,a+\frac{h}{2}\right]$和$\left[a+\frac{h}{2},a+h\right]$,区间长度为$\frac{h}{2}$.在这两个 1 级子区间上应用辛普森求积公式(4.2.13),得到结果 $S^{(1)}_{a,a+\frac{h}{2}}$和 $S^{(1)}_{a+\frac{h}{2},a+h}$.然后在 1 级区间上计算

$$S^{(2)}_{a,a+\frac{h}{2}}=S^{(1)}_{a,a+\frac{h}{2^2}}+S^{(1)}_{a+\frac{h}{2^2},a+\frac{h}{2}}.$$

在 1 级区间$\left[a,a+\frac{h}{2}\right]$上比较 $S^{(1)}_{a,a+\frac{h}{2}}$与 $S^{(2)}_{a,a+\frac{h}{2}}$,如果不等式

$$|S^{(1)}_{a,a+\frac{h}{2}}-S^{(2)}_{a,a+\frac{h}{2}}|<10\times\frac{\varepsilon}{2} \qquad (4.9.7)$$

成立,则说明在 1 级子区间$\left[a,a+\frac{h}{2}\right]$上已达到要求,然后在下一个 1 级子区间$\left[a+\frac{h}{2},a+h\right]$上计算

$$S^{(2)}_{a+\frac{h}{2},a+h}=S^{(1)}_{a+\frac{h}{2},a+\frac{1}{2^2}3h}+S^{(1)}_{a+\frac{1}{2^2}\cdot 3h,a+h,a+h}.$$

如果不等式

$$|S^{(2)}_{a+\frac{h}{2},a+h}-S^{(1)}_{a+\frac{h}{2},a+h}|<10\times\frac{\varepsilon}{2} \qquad (4.9.8)$$

成立，则认为在整个区间$[a,a+h]$上计算完成，令

$$S_{a,a+h}=S^{(2)}_{a,a+\frac{h}{2}}+S^{(2)}_{a+\frac{h}{2},a+h},$$

这就是$I(f)$的近似值.

如果不等式(4.9.7)和(4.9.8)中有一个不成立，例如式(4.9.7)不成立，则不考虑1级子区间$\left[a+\frac{h}{2},a+h\right]$，而将$\left[a,a+\frac{h}{2}\right]$分成两个相等的2级子区间$\left[a,a+\frac{h}{2^2}\right]$和$\left[a+\frac{h}{2^2},a+\frac{h}{2}\right]$. 对2级子区间$\left[a,a+\frac{h}{2^2}\right]$采用辛普森公式(4.2.9)计算$S^{(1)}_{a,a+\frac{1}{2^2}h}$和$S^{(2)}_{a,a+\frac{1}{2^2}h}=S^{(1)}_{a,a+\frac{1}{2^3}h}+S^{(1)}_{a+\frac{1}{2^3}h,a+\frac{1}{2^2}h}$，然后比较$S^{(1)}_{a,a+\frac{1}{2^2}h}$和$S^{(2)}_{a,a+\frac{1}{2^2}h}$，如果不等式

$$\left|S^{(1)}_{a,a+\frac{1}{2^2}h}-S^{(2)}_{a,a+\frac{1}{2^2}h}\right|<10\times\frac{\varepsilon}{2^2} \tag{4.9.9}$$

满足，则说明在2级子区间$\left[a,a+\frac{1}{2^2h}\right]$上计算已达到要求，继续在其右边2级子区间$\left[a+\frac{1}{2^2}h,a+\frac{1}{2}h\right]$上进行类似计算，并满足类似于式(4.9.9)的不等式，此时可取

$$S_{a,a+h}=S^{(2)}_{a,a+\frac{1}{2^2}h}+S^{(2)}_{a+\frac{1}{2^2}h,a+\frac{1}{2}h}+S^{(2)}_{a+\frac{1}{2}h,a+h}$$

为$I(f)$的近似值.

如果不等式(4.9.9)不成立，则再进一步将2级子区间$\left[a,a+\frac{h}{2^2}\right]$分成两个相等的3级子区间等，类似上述步骤继续进行.

例 4.9.1 用自适应辛普森求积公式计算$I(f)=\int_1^3 f(x)\mathrm{d}x$，其中$f(x)=\frac{100}{x^2}\sin\frac{10}{x}$.

解 自适应辛普森求积方法中允许误差为 10^{-4}. 方法要求 23 个小区间. $f(x)$的图形及小区间的端点分布见图 4.3. 计算有

$$\int_{1}^{3} f(x)\,\mathrm{d}x \approx -1.426014,$$

此近似值已在 1.1×10^{-5} 的精度范围之内,其中计算函数值的总数为 93.

若用复合辛普森求积公式使其误差小于等于 10^{-4}, h 的取值为$\frac{1}{88}$,需计算 177 次函数值. 由此看出,接近于自适应辛普森求和方法的两倍.

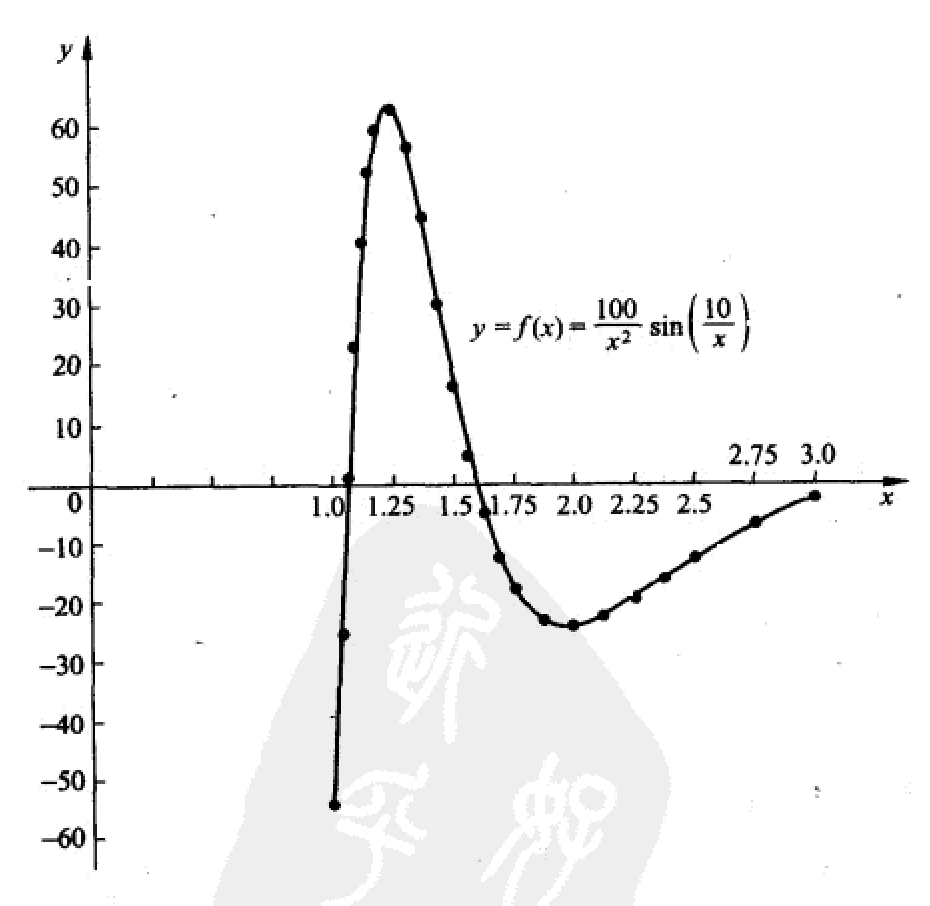


图 4.3

4.10 奇异积分的计算

奇异积分(singular integral)的计算要针对具体情况选择具体方法.

4.10.1 积分变量替换

采用积分变量替换(change of the variable of integration)可消除奇异性,举例说明如下.

例 4.10.1 设 $f\in C[a,b]$,计算奇异积分

$$\int_0^1 x^{-\frac{1}{n}} f(x)\mathrm{d}x,\quad n\geqslant 2.$$

解 为消去奇异性,令 $u^n=x$,那么将积分化为

$$n\int_0^1 f(u^n)u^{n-2}\mathrm{d}u,$$

这是一个正常积分.

例 4.10.2 考虑积分

$$I(f)=\int_0^1 \sin x\sqrt{1-x^2}\mathrm{d}x.$$

解 被积函数的第 2 个因子$\sqrt{1-x^2}$的一阶导数在 $x=0$ 处有奇点.令

$$u=\sqrt{1-x},$$

那么有

$$I(f)=2\int_0^1 u^2\sqrt{2-u^2}\sin(1-u^2)\mathrm{d}u.$$

例 4.10.3 讨论

$$I(f)=\int_0^a x^{\frac{p}{q}} f(x)\mathrm{d}x,$$

其中$\frac{p}{q}$为既约分数,$f\in C[0,a]$.

解 令 $x=u^q$,可得

$$I(f)=q\int_0^{a^{\frac{1}{q}}} u^{p+q-1}f(u^q)\mathrm{d}u.$$

例 4.10.4 常用的变量替换有

$$\int_{-1}^{1}\frac{f(x)}{\sqrt{1-x^2}}\mathrm{d}x=\int_0^{\pi}f(\cos u)\mathrm{d}u,$$

$$\int_{-1}^{1}\frac{f(x)}{\sqrt{x(1-x)}}\mathrm{d}x=2\int_0^{\frac{\pi}{2}}f(\sin^2 u)\mathrm{d}u.$$

上述第一个积分可使用高斯-切比雪夫求积公式(4.5.8).

应注意到,有时采用变量替换可能没有解决问题的困难.例如,对于积分$\int_0^1 f(x)\ln x\mathrm{d}x$, $f\in C[a,b]$, $f(0)\neq 0$,若用变换$u=-\ln x$,那么就得到$-\int_0^{\infty}u\mathrm{e}^{-u}f(\mathrm{e}^{-u})\mathrm{d}u$,这是一个无穷区间上的积分,因此困难没有解决.

4.10.2 极限过程

设 $f(x)$在 $x=0$ 的邻域内无界,反常积分可以定义为

$$\int_0^1 f(x)\mathrm{d}x=\lim_{r\to 0}\int_r^1 f(x)\mathrm{d}x, \tag{4.10.1}$$

由此可得到一个计算方案.令 $1>r_1>r_2>\cdots$是收敛于 0 的数列,例如 $r_n=2^{-n}$.记

$$\int_0^1 f(x)\mathrm{d}x=\int_{r_1}^1 f(x)\mathrm{d}x+\int_{r_2}^{r_1}f(x)\mathrm{d}x+\int_{r_3}^{r_2}f(x)\mathrm{d}x+\cdots$$

右边的每个积分都是正常积分.一般地,当$\left|\int_{r_{n+1}}^{r_n}f(x)\mathrm{d}x\right|<\varepsilon$时,计算停止.

必须注意,这仅是一种实践准则.

例 4.10.5 计算积分

$$I(f)=\int_0^1 \frac{1}{x^{\frac{1}{2}}+x^{\frac{1}{3}}}\mathrm{d}x.$$

解 取 $r_n=2^{-n}$，计算 $I_n=\int_{r_n}^1 \frac{1}{x^{\frac{1}{2}}+x^{\frac{1}{3}}}\mathrm{d}x$，其结果见表 4.14.

表 4.14

n	I_n
1	0.28492598
2	0.47448022
4	0.68323927
8	0.81280497
16	0.84029678
32	0.84111612
40	0.8411663
精确值	0.8411692

4.10.3 奇异性的解析处理

奇异性的解析处理(analytic treatment of singularity)也称区间的截去方法，是把积分区间分成两部分，使一部分有奇异点而另一部分没有奇异点. 如果

$$I=\int_a^b f(x)\mathrm{d}x$$

中被积函数 f 在 $x=a$ 处有奇异点，则适当地选取小数 $\delta>0$，可使在小区间 $[a,a+\delta]$ 上的积分值处在允许的误差范围之内，即

$$\left|\int_a^{a+\delta} f(x)\mathrm{d}x\right|<\varepsilon.$$

而对于积分

$$\int_{a+\delta}^b f(x)\mathrm{d}x,$$

则可以按标准的数值积分方法进行.

例 4.10.6 计算积分

$$\int_0^1 \frac{g(x)}{x^{\frac{1}{2}}+x^{\frac{1}{3}}}\mathrm{d}x,$$

其中 g 在[0,1]上充分光滑，且满足 $|g(x)|\leqslant 1, x\in[0,1]$.

解 因为在[0,1]上，$x^{\frac{1}{2}}\leqslant x^{\frac{1}{3}}$，则有

$$\left|\frac{g(x)}{x^{\frac{1}{2}}+x^{\frac{1}{3}}}\right|\leqslant\frac{1}{2x^{\frac{1}{2}}},$$

因此

$$\left|\int_0^\delta \frac{g(x)}{x^{\frac{1}{2}}+x^{\frac{1}{3}}}\mathrm{d}x\right|\leqslant\frac{1}{2}\int_0^\delta\frac{1}{x^{\frac{1}{2}}}\mathrm{d}x=\delta^{\frac{1}{2}}.$$

如果精度要求为 10^{-3}，则 $\delta\leqslant 10^{-6}$，而在$[\delta,1]$上的积分

$$\int_\delta^1 \frac{g(x)}{x^{\frac{1}{2}}+x^{\frac{1}{3}}}\mathrm{d}x,$$

可以采用标准方法进行计算.

例 4.10.7 计算积分

$$I=\int_0^b f(x)\ln x\mathrm{d}x.$$

解 先将积分 I 分成两部分：

$$I=\int_0^\delta f(x)\ln x\mathrm{d}x+\int_\delta^b f(x)\ln x\mathrm{d}x\equiv I_1+I_2.$$

假定 $f(x)$在$[\delta,b]$上充分光滑，则可用标准方法数值计算 I_2. 假定 $f(x)$在$[0,\epsilon]$上可展成收敛的泰勒级数，并使用初等积分

$$\begin{aligned}\int_0^\delta x^i\ln x\mathrm{d}x&=\frac{x^{i+1}}{i+1}\left(\ln x-\frac{1}{i+1}\right)\Big|_0^\delta\\&=\frac{\delta^{i+1}}{i+1}\left(\ln\delta-\frac{1}{i+1}\right).\end{aligned}$$

这样可得到

$$I_1=\int_0^\delta f(x)\ln x\mathrm{d}x=\int_0^\delta\left(\sum_{i=0}^{\infty}a_ix^i\right)\ln x\mathrm{d}x$$

$$= \sum_{i=0}^{\infty} a_i \frac{\delta^{i+1}}{i+1}\left(\ln\delta - \frac{1}{i+1}\right).$$

对于给定的精度,可以对级数进行估计.

取 $f(x)=\cos x, b=4\pi$,即计算

$$I = \int_0^{4\pi} \cos x \ln x \mathrm{d}x.$$

取 $\delta=0.1$,则

$$\begin{aligned} I_1 &= \int_0^{0.1} \cos x \ln x \mathrm{d}x \\ &= \delta(\ln\delta - 1) - \frac{\delta^3}{6}\left(\ln\delta - \frac{1}{3}\right) + \frac{\delta^5}{600}\left(\ln\delta - \frac{1}{5}\right) - \cdots \end{aligned}$$

这是一个交错级数,取前三项可以求得 I_1 的相当精确的值,而在 $[0.1, 4\pi]$ 上的积分 I_2 可以用标准的方法求出.

4.10.4 乘积积分

考虑积分

$$I(f) = \int_a^b w(x) f(x) \mathrm{d}x,$$

其中 w 是一个奇异的权函数, f 是一个光滑的函数. 构造一个函数序列 f_n,使得

(1) 当 $n\to\infty$ 时有

$$\| f - f_n \|_\infty = \max_{a\leqslant x\leqslant b} | f(x) - f_n(x) | \to 0.$$

(2) 积分

$$I_n(f) \equiv \int_a^b w(x) f_n(x) \mathrm{d}x$$

容易计算.

乘积积分方法(product integration method)通常采用 f 的分段多项式插值 f_n 来确定 $I_n(f)$,可以用乘积梯形方法(product trapezoidal method)来计算积分

$$I(f)=\int_0^b f(x)\ln x\mathrm{d}x. \tag{4.10.2}$$

令 $n\geqslant 1, h=\frac{b}{n}, x_j=jh(j=0,1,\cdots,n)$，$f_n$ 定义为 f 在节点 $x_0,x_1,\cdots,x_n$ 上的分段线性函数插值. 对于 $j=1,2,\cdots,n$，定义

$$f_n(x)=\frac{1}{h}[(x_j-x)f(x_{j-1})+(x-x_{j-1})f(x_j)]$$

$$(x_{j-1}\leqslant x\leqslant x_j) \tag{4.10.3}$$

如果 f 在$[a,b]$上二次连续可微，那么利用插值多项式的误差估计可以得到

$$\| f-f_n\|_\infty\leqslant\frac{h^2}{8}\| f''\|_\infty.$$

由此有

$$| I(f)-I_n(f) |\leqslant\frac{h^2}{8}\| f''\|_\infty\int_0^b | \ln x | \mathrm{d}x.$$

而 $I_n(f)$是容易计算的，

$$\begin{aligned}I_n(f)&=\sum_{j=1}^n\int_{x_{j-1}}^{x_j}(\ln x)\left[\frac{(x_j-x)f(x_{j-1})+(x-x_{j-1})f(x_j)}{h}\right]\mathrm{d}x\\&=\sum_{k=0}^n A_k f(x_k),\end{aligned} \tag{4.10.4}$$

其中

$$\begin{cases}A_0=\frac{1}{h}\int_{x_0}^{x_1}(x_1-x)\ln x\mathrm{d}x,\\A_n=\frac{1}{h}\int_{x_{n-1}}^{x_n}(x-x_{n-1})\ln x\mathrm{d}x,\\A_j=\frac{1}{h}\int_{x_{j-1}}^{x_j}(x-x_{j-1})\ln x\mathrm{d}x+\frac{1}{h}\int_{x_j}^{x_{j+1}}(x_{j+1}-x)\ln x\mathrm{d}x,\\\qquad j=1,\cdots,n-1.\end{cases}$$

作变量替换 $x-x_{j-1}=uh(0\leqslant u\leqslant 1)$，有

$$\frac{1}{h}\int_{x_{j-1}}^{x_j}(x-x_{j-1})\ln x\mathrm{d}x=\frac{h}{2}\ln h+h\int_0^1 u\ln(j-1+u)\mathrm{d}u$$

和

$$\frac{1}{h}\int_{x_{j-1}}^{x_j}(x_j-x)\ln x\mathrm{d}x=\frac{h}{2}\ln h+h\int_0^1(1-u)\ln(j-1+u)\mathrm{d}u.$$

令

$$\psi_1(k)=\int_0^1 u\ln(u+k)\mathrm{d}u, \tag{4.10.5}$$

$$\psi_2(k)=\int_0^1(1-u)\ln(u+k)\mathrm{d}u \quad (k=0,1,\cdots,n),$$

可得

$$\begin{cases}w_0=\dfrac{h}{2}\ln h+h\psi_2(0),\\ w_n=\dfrac{h}{2}\ln h+h\psi_1(n-1),\\ w_j=h\ln h+h[\psi_1(j-1)+\psi_2(j)] \quad (j=1,2,\cdots,n-1),\end{cases} \tag{4.10.6}$$

$\psi_1(k)$和$\psi_2(k)$不依赖于h,b或n,因此它们可以预先计算好.$\psi_1(k),\psi_2(k)$的前8个数值见表4.15.

表 4.15

k	$\psi_1(k)$	$\psi_2(k)$
0	−0.250	−0.75
1	0.250	0.1362943611
2	0.4883759281	0.4211665768
3	0.6485778545	0.6007627239
4	0.7695705457	0.7324415720
5	0.8668602747	0.8365069785
6	0.9482428376	0.9225713904
7	1.018201652	0.9959596385

4.10.5 削减奇异性方法

把积分 $I(f)=\int_a^b f(x)\mathrm{d}x$ 分解为奇异和非奇异两部分，奇异部分可用解析方法求解，非奇异部分可应用标准数值方法求解. 此方法称为**削减奇异性方法**(methods of subtracting the singularity).

例 4.10.8 计算奇异积分

$$I=\int_0^1 \frac{\mathrm{e}^x}{\sqrt{x}}\mathrm{d}x,$$

精确到 3 位.

解 分解积分为

$$I=\int_0^1 \frac{1}{\sqrt{x}}\mathrm{d}x+\int_0^1 \frac{\mathrm{e}^x-1}{\sqrt{x}}\mathrm{d}x.$$

第一个积分是初等积分，而第二个积分无奇异点. 注意到由于 $\frac{1}{\sqrt{x}}(\mathrm{e}^x-1)$ 在 0 附近性态与 $\sqrt{x}$ 相似，它有一个奇异点在其一阶导数中，因而使数值积分不精确，为此将积分写成

$$\begin{aligned}I&=\int_0^1 \frac{1+x}{\sqrt{x}}\mathrm{d}x+\int_0^1 \frac{\mathrm{e}^x-1-x}{\sqrt{x}}\mathrm{d}x\\&\approx\frac{8}{3}+0.2583\approx 2.925.\end{aligned}$$

4.10.6 康托洛维奇方法

积分

$$I(f)=\int_a^b f(x)\mathrm{d}x$$

的被积函数 f 存在一个奇异点. 康托洛维奇(Kontorovich)方法不是直接对 $I(f)$ 进行求积. 而是选取一个函数 g，使其与 f 有相同的

奇异点，并在给定的积分区间$[a,b]$上可解析求积，而且 $f-g$ 有一定阶的导数．把积分写成

$$\int_a^b f(x)\mathrm{d}x=\int_a^b g(x)\mathrm{d}x+\int_a^b[f(x)-g(x)]\mathrm{d}x,$$

右边第一个积分可直接求积，第二个积分可用标准的数值求积公式计算．

康托洛维奇方法应属于削减奇异性方法范畴．

函数 g 的选取，有很多方法．例如被积函数 f 用公式

$$f(x)=(x-c)^{\alpha}\varphi(x),\quad a\leqslant c\leqslant b,\quad x\in[a,b] \tag{4.10.7}$$

来表示，其中$-1<\alpha<0$，φ 在$[a,b]$上足够光滑．φ 在 $x=c$ 处展成泰勒级数，则可以得到

$$\begin{aligned}f(x)=&\left[\varphi(c)(x-c)^{\alpha}+\frac{\varphi'(c)}{1!}(x-c)^{\alpha+1}+\right.\\&\left.\frac{\varphi''(c)}{2!}(x-c)^{\alpha+2}+\cdots+\frac{\varphi^{(k)}(c)}{k!}(x-c)^{\alpha+k}\right]+\\&(x-c)^{\alpha}\left[\varphi(x)-\varphi(c)-\frac{\varphi'(c)}{1!}(x-c)-\right.\\&\left.\frac{\varphi''(c)}{2!}(x-c)^{2}-\cdots-\frac{\varphi^{(k)}(c)}{k!}(x-c)^{k}\right].\end{aligned} \tag{4.10.8}$$

上式右边第一个方括号中是一个幂函数，可以逐项求其积分；而第二个方括号内已无奇点，且相当光滑，可以用标准的数值求积公式计算出来．

例 4.10.9 求 $I=\int_0^{\frac{1}{2}}\frac{1}{\sqrt{x(1-x)}}\mathrm{d}x$ 的近似值．

解 被积函数在 $x=0$ 处间断，且可用公式

$$f(x)=x^{-\frac{1}{2}}(1-x)^{-\frac{1}{2}}$$

表示，于是 $\alpha=-\frac{1}{2}, c=0, \varphi(x)=(1-x)^{-\frac{1}{2}}$. 由泰勒级数展开，有

$$\varphi(x)=1+\frac{1}{2}x+\frac{3}{8}x^2+\frac{5}{16}x^3+\frac{35}{128}x^4+R_4(x).$$

因此，f 可以写成

$$f(x)=\left(x^{-\frac{1}{2}}+\frac{1}{2}x^{\frac{1}{2}}+\frac{3}{8}x^{\frac{3}{2}}+\frac{5}{16}x^{\frac{5}{2}}+\frac{35}{128}x^{\frac{7}{2}}\right)+\frac{\psi(x)}{\sqrt{x}},$$

其中

$$\psi(x)=\frac{1}{\sqrt{1-x}}-\left(1+\frac{1}{2}x+\frac{3}{8}x^2+\frac{5}{16}x^3+\frac{35}{128}x^4\right).$$

因此

$$\begin{aligned}I&=\int_0^{\frac{1}{2}}\left(x^{-\frac{1}{2}}+\frac{1}{2}x^{\frac{1}{2}}+\frac{3}{8}x^{\frac{3}{2}}+\frac{5}{16}x^{\frac{5}{2}}+\frac{35}{128}x^{\frac{7}{2}}\right)\mathrm{d}x+I_1\\&=1.5691585+I_1,\end{aligned}$$

其中

$$I_1=\int_0^{\frac{1}{2}}\frac{\psi(x)}{\sqrt{x}}\mathrm{d}x.$$

用 $n=10$ 的复合辛普森公式计算 I_1，得 $I_1=0.0016385$，由此得$I=1.5707970$.

为比较起见，求出 I 的精确值，$I=\frac{\pi}{2}=1.5707963$. 由此可见是相当精确的.

4.10.7 高斯求积

如果被积函数分解成两个函数的乘积，那么奇异积分常可应用高斯型求积，考虑

$$I(f)=\int_a^b w(x)f(x)\mathrm{d}x, \tag{4.10.9}$$

其中 w 是一个固定的非负的权函数. 它在积分区间$[a,b]$上存在

一个或几个奇异点，并且积分$\int_a^b w(x)x^k\mathrm{d}x(k=0,1,\cdots,n)$存在，而 f 是一个充分光滑的函数. 这样可以按 4.5 节的方法来确定积分公式的节点和系数. w 有以下几种特殊情况：

(1) $w(x)=(1-x^2)^{\frac{1}{2}}$，相应的求积公式

$$\int_{-1}^{1}\sqrt{1-x^2}f(x)\mathrm{d}x\approx\sum_{k=1}^{n}A_kf(x_k),\qquad(4.10.10)$$

其中

$$x_k=\cos\frac{k+1}{n+1}\pi,\quad A_k=\frac{\pi}{n+1}\sin^2\frac{k+1}{n+1}\pi,$$

误差为

$$E_n=\frac{\pi}{(2n)!2^{2n+1}}f^{(2n)}(\eta),\quad \eta\in(-1,1).$$

(2) $w(x)=\sqrt{x}\cdot\dfrac{1}{\sqrt{1-x}}$，相应的求积公式

$$\int_0^1\sqrt{\frac{x}{1-x}}f(x)\mathrm{d}x\approx\sum_{k=1}^{n}A_kf(x_k),\qquad(4.10.11)$$

其中

$$x_k=\cos^2\frac{2k-1}{2n+1}\frac{\pi}{2},\quad A_k=\frac{2\pi}{2n+1}x_k,$$

误差为

$$E_n(f)=\frac{\pi}{(2n)!2^{4n+1}}f^{(2n)}(\eta),\quad \eta\in(0,1).$$

(3) $w(x)=(1-x)^{\frac{1}{2}}$，相应的求积公式

$$\int_0^1\sqrt{1-x}f(x)\mathrm{d}x\approx\sum_{k=1}^{n}A_kf(x_k),\qquad(4.10.12)$$

其中

$$x_k=1-z_k^2$$

（z_k 是 $2n+1$ 次勒让德多项式 P_{2n+1} 的第 k 个正零点），

$$A_k = 2z_k^2 A_k^{(2n+1)},$$

此处 $A_k^{(2n+1)} = \dfrac{2}{(1-z_k^2)[\mathrm{P}'_{2n+1}(z_k)]^2}$ 是 $2n+1$ 个节点的高斯-勒让德求积公式中相应于节点 z_k 的系数.

误差为

$$E_n(f) = \frac{2^{4n+3}[(2n+1)!]^4}{(2n)!(4n+3)[(4n+2)!]^2} f^{(2n)}(\eta), \quad \eta \in (0,1).$$

(4) $w(x) = (1-x)^{-\frac{1}{2}}$,相应的求积公式

$$\int_0^1 \frac{f(x)}{\sqrt{1-x}} \mathrm{d}x \approx \sum_{k=1}^n A_k f(x_k), \tag{4.10.13}$$

其中 $x_k = 1 - z_k^2$(z_k 是 $2n$ 次勒让德多项式 $\mathrm{P}_{2n}(x)$ 的第 k 个正零点),

$$A_k = 2A_k^{(2n)},$$

此处 $A_k^{(2n)} = \dfrac{2}{(1-z_k^2)[\mathrm{P}'_{2n}(z_k)]^2}$ 是 $2n$ 个节点的高斯-勒让德求积公式中相应于节点 z_k 的系数.

误差为

$$E_n(f) = \frac{2^{4n+1}[(2n)!]^3}{(4n+1)[(4n)!]^2} f^{(2n)}(\eta), \quad \eta \in (0,1).$$

(5) $w = (1-x^2)^{-\frac{1}{2}}$,相应的求积公式为高斯-切比雪夫公式(4.5.8).

4.11 振荡函数的积分

$$I(t) = \int_a^b f(x) K(x,t) \mathrm{d}x, \tag{4.11.1}$$

其中 $K(x,t)$ 是一个"振荡核",即 $K(x,t)$ 是关于 x 的振荡函数.而

f 为非振荡函数. 傅里叶积分

$$\int_a^b f(x)\sin nx\,\mathrm{d}x,\quad \int_a^b f(x)\cos nx\,\mathrm{d}x$$

为典型例子.

4.11.1 在两零点之间的积分

设被积函数振荡部分在$[a,b]$上的零点为 $a\leqslant x_1<x_2<\cdots<x_p\leqslant b$,那么将$[a,b]$上的积分分解为子区间$[x_k,x_{k+1}]$上积分之和. 在每个子区间上,因被积函数在端点之值为零,所以可以采用高斯-洛巴托求积公式(4.6.2),这样可以不必增加计算量就可以得到较高的精度. 例如

$$\int_0^{2\pi} f(x)\sin nx\,\mathrm{d}x=\sum_{k=0}^{2n-1}\int_{\frac{k\pi}{n}}^{\frac{(k+1)\pi}{n}} f(x)\sin nx\,\mathrm{d}x,$$

等式右边求和号中每项积分的被积函数在端点处的值均为零,因此$\int_{\frac{k\pi}{n}}^{\frac{(k+1)\pi}{n}} f(x)\sin nx\,\mathrm{d}x$ 可用高斯-洛巴托求积公式计算,若用 5 个节点的高斯-洛巴托求积公式(2 个端点,3 个内点),那么在每个子区间上只要计算 3 个函数值就可以了.

对于$\int_0^{2\pi} f(x)\cos nx\,\mathrm{d}x$ 可用类似的方法计算.

4.11.2 菲隆方法

设 f 可以表示为

$$f(x)=\sum_{k=1}^{n} a_k\varphi_k(x)+\varepsilon(x),\quad x\in[a,b],\quad (4.11.2)$$

其中 $\varepsilon(x)$是$[a,b]$上的小量(当然为 x 的函数). 令

$$\psi_k(t)=\int_a^b \varphi_k(x)K(x,t)\mathrm{d}x\quad (k=1,2,\cdots,n)$$

可以用初等积分显式表示出来. 如当 $\varphi_k(x)=x^k$, $K(x,t)=e^{itx}$ 时就是这种情况.

$$I(t)=\int_a^b f(x)K(x,t)\mathrm{d}x=\sum_{k=1}^n a_k\psi_k(t)+\int_a^b \varepsilon(x)K(x,t)\mathrm{d}x$$

$$\approx\sum_{k=1}^n a_k\psi_k(t). \tag{4.11.3}$$

上述算法是由菲隆(Filon)首先给出的,因此称为**菲隆算法**.

考虑积分

$$I(k)=\int_a^b f(x)\sin kx\,\mathrm{d}x.$$

把区间$[a,b]$等分为 $2N$ 个子区间,在每两个子区间上用 f 的二次插值多项式来代替 f,那么相应的积分可以用分部积分准确计算,于是

$$\int_a^b f(x)\sin kx\,\mathrm{d}x\approx h\{-\alpha[f(b)\cos kb-f(a)\cos ka]+$$

$$\beta S_{2N}+rS_{2N-1}\}, \tag{4.11.4}$$

其中

$$h=\frac{b-a}{2N},$$

$$\begin{cases}\alpha=\alpha(\theta)=\dfrac{(\theta^2+\theta\sin\theta\cos\theta-2\sin^2\theta)}{\theta^3},\\ \beta=\beta(\theta)=\dfrac{2[\theta(1+\cos^2\theta)-2\sin\theta\cos\theta]}{\theta^3},\\ \gamma=\gamma(\theta)=\dfrac{4(\sin\theta-\theta\cos\theta)}{\theta^3},\end{cases} \tag{4.11.5}$$

$$\theta=kh,$$

$$S_{2N}=\frac{1}{2}f(a)\sin ka+f(a+2h)\sin k(a+2h)+$$

$$f(a+4h)\sin k(a+4h)+\cdots+\frac{1}{2}f(b)\sin kb, \tag{4.11.6}$$

$$S_{2N-1}=f(a+h)\sin k(a+h)+f(a+3h)\sin k(a+3h)+\cdots+$$
$$f(b-h)\sin(b-h). \tag{4.11.7}$$

类似地，有

$$\int_a^b f(x)\cos kx\,\mathrm{d}x \approx h\{\alpha[f(b)\sin kb-f(a)\sin ka]+$$
$$\beta C_{2N}+\gamma C_{2N-1}\}, \tag{4.11.8}$$

其中 C_{2N-1}，C_{2N} 与 S_{2N-1}，S_{2N} 相对应，用相应的 $\cos\zeta$ 来代替 $\sin\zeta$ 即可.

应注意，当 θ 较小时，函数 α,β,γ 有泰勒级数展开

$$\begin{cases}\alpha(\theta)=\dfrac{2}{45}\theta^3-\dfrac{2}{315}\theta^5+\dfrac{2}{4725}\theta^7+\cdots\\ \beta(\theta)=\dfrac{2}{3}+\dfrac{2}{15}\theta^2+\dfrac{4}{105}\theta^4+\dfrac{2}{567}\theta^6+\cdots\\ \gamma(\theta)=\dfrac{4}{3}-\dfrac{2}{15}\theta^2+\dfrac{1}{210}\theta^4-\dfrac{1}{11340}\theta^6+\cdots\end{cases} \tag{4.11.9}$$

例 4.11.1 用 Filon 方法计算积分

$$I(k)=\int_0^{2\pi} f(x)\sin kx\,\mathrm{d}x,$$

其中 $f(x)=x\cos x$.

解 $$I(k)=\int_0^{2\pi} x\cos x\sin kx\,\mathrm{d}x=\begin{cases}-\dfrac{\pi}{2}, & k=1,\\ -\dfrac{2k\pi}{k^2-1}, & k\neq 1.\end{cases}$$

用 G_{32} 表示 32 个节点的高斯求积公式，用 $2k\times L_4$ 与 $2k\times L_5$ 分别表示以 $\sin kx$ 的零点划分子区间（共 $2k$ 个）上四点与五点的高斯-洛巴托求积公式，F_i 表示 $h=2\pi/ik$ 的菲隆方法(4.11.4). 计算结果见表 4.16.

表 4.16

k	准确值	G_{32}	$2k\times L_4$	$2k\times L_5$
10	−0.63466518	−0.63402096	−0.70206954	−0.5587594
20	−0.31494663	−1.2092524	−0.34818404	−0.2778962
30	−0.20967243	−1.5822272	−0.23177723	−0.18508448

k	F_7	F_8	F_{11}
10	−0.63466469	−0.63466497	−0.63466508
20	−0.31494462	−0.31494463	−0.31494463
30	−0.20967248	−0.20967248	−0.20967248

注意到，当 k 越大时，振荡越严重，高斯型求积公式精度很差. 但对于小的 k，高阶高斯求积公式还较好，例如，当 $n=1$ 时，积分准确值为 −1.57079633，$G_{32}=-1.5704811$；当 $n=2$ 时，积分准确值为 −4.1887902，$G_{32}=-4.1842807$.

4.12 无穷区间上的积分

考虑无穷区间上的积分

$$I(f)=\int_a^{\infty} f(x)\mathrm{d}x, \tag{4.12.1}$$

其中 a 为有限值或 $-\infty$.

4.12.1 变量替换

对于式(4.12.1)，作变量替换 $t=\mathrm{e}^{-x}$，可将区间 $[0,+\infty)$ 变为区间 $(0,1)$. 因此有

$$\int_0^{\infty} f(x)\mathrm{d}x=\int_0^1 \frac{1}{t}f(-\ln t)\mathrm{d}t=\int_0^1 \frac{g(t)}{t}\mathrm{d}t. \tag{4.12.2}$$

这样就把无穷区间上的一个积分化成了有限区间上的积分. 若 $g(t)/t$ 在 $t=0$ 的邻域内有界，那么式(4.12.2)的右边是一个正常

积分，反之，积分是一个反常积分，上述变换只是把一种困难转换成另一种困难.

变量替换还有许多不同类型.

例 4.12.1 计算积分

$$\int_1^{\infty} \frac{1}{x^2}\sin\frac{1}{x^2}\mathrm{d}x.$$

解 令 $y=\frac{1}{x}$，那么有

$$\int_1^{\infty} \frac{1}{x^2}\sin\frac{1}{x^2}\mathrm{d}x = \int_0^1 \sin y^2\mathrm{d}y,$$

对 $\sin y^2$ 泰勒级数展开，有

$$\int_0^1 \sin y^2\mathrm{d}y = \frac{1}{3}-\frac{1}{42}+\frac{1}{1320}-\frac{1}{75600}+\cdots$$
$$\approx 0.310268.$$

4.12.2 无穷区间的截断

将被积函数的"尾巴"略去，可使无穷区间化为一个有限区间. 此方法要求事先用某种简单的解析方法估算出尾部的量值. 选取 $R>a$，使

$$\left|\int_R^{\infty} f(x)\mathrm{d}x\right| < \varepsilon,$$

其中 ε 为允许误差，那么无穷区间上的积分(4.12.1)可以用 $\int_a^R f(x)\mathrm{d}x$ 来近似.

例 4.12.2 计算

$$\int_0^{\infty} \mathrm{e}^{-x^2}\mathrm{d}x.$$

解 当 $x\geqslant R$ 时有 $x^2\geqslant Rx$，所以有估计式

$$\int_R^{\infty} \mathrm{e}^{-x^2}\mathrm{d}x \leqslant \int_R^{\infty} \mathrm{e}^{-Rx}\mathrm{d}x = \frac{1}{R}\mathrm{e}^{-R^2}.$$

对于 $R=4$，则$\frac{1}{R}e^{-R^2}\approx 10^{-8}$. 因此对于允许误差为 10^{-7} 来说，只要计算 $\int_0^4 e^{-x^2}dx$ 就可以了.

例 4.12.3 计算

$$\int_0^\infty \frac{\sin x}{1+x^2}dx.$$

解 令

$$r_j=\int_{(2k+j-1)\pi}^{(2k+j)\pi}\frac{\sin x}{1+x^2}dx,$$

那么，有

$$\left|\int_{2k\pi}^\infty \frac{\sin x}{1+x^2}dx\right|=|r_1+r_2+\cdots|.$$

由于 $r_{2n}<0, r_{2n+1}>0$，以及 $|r_1|>|r_2|>\cdots$，所以有

$$|r_1+r_2+\cdots|<r_1=\int_{2k\pi}^{(2k+1)\pi}\frac{\sin x}{1+x^2}dx$$

$$<\int_{2k\pi}^{(2k+1)\pi}\frac{1}{x^2}dx<\frac{1}{4\pi k^2}.$$

若截断误差为 10^{-4}，可取 $k\approx 28$.

4.12.3 无穷区间上的高斯求积公式

无穷区间上的积分. 高斯-拉盖尔求积公式(4.5.9)和高斯-埃尔米特求积公式(4.5.10)是使用最广泛的. 下面作些补充.

(1) 移位的拉盖尔公式

对于积分

$$\int_a^\infty e^{-t}f(t)dt,$$

作变量替换 $t=x+a$，则得

$$\int_a^\infty e^{-t}f(t)dt=e^{-a}\int_0^\infty e^{-x}f(x+a)dx,$$

右端每个积分都是正常积分，当

$$\left|\int_{r_n}^{r_{n+1}} f(x)\,\mathrm{d}x\right| < \varepsilon$$

时，计算终止.

例 4.12.4 计算积分

$$I = \int_0^{\infty} \frac{\mathrm{e}^{-x}}{1+x^4}\mathrm{d}x.$$

解 取 $r_n = 2^n$，$I_n = \int_0^{r_n} \frac{\mathrm{e}^{-x}}{1+x^4}\mathrm{d}x$，$I_n$ 计算结果见表 4.17.

表 4.17

n	I_n	计算函数值个数
0	0.57202582	35
1	0.62745952	52
2	0.63043990	100
3	0.63047761	178
4	0.63047766	322
精确值	0.63047783	

如果能找到 $\int_r^{\infty} f(x)\,\mathrm{d}x$ 的一个较合理而良好的估计，那么可以进行外推加速.

用移位的拉盖尔求积公式(4.12.3)来计算例 4.12.4 中积分

$$\int_r^{\infty} \frac{\mathrm{e}^{-x}}{1+x^4}\mathrm{d}x \approx c\,\frac{\mathrm{e}^{-r}}{1+r^4}.$$

用里查森外推法有

$$I'_n = \frac{I_n\phi(r_{n+1}) - I_{n+1}\phi(r_n)}{\phi(r_{n+1}) - \phi(r_n)},$$

$$\phi(r) = \frac{\mathrm{e}^{-r}}{1+r^4}, \quad r_n = 2^n,$$

计算结果见表 4.18.

表 4.18

n	I'_n
0	0.62996722
1	0.63046682
2	0.63047765
3	0.63047766

注意到，I'_1 比 I_2 好，I'_2 与 I_4 几乎相等.

4.13 重积分的数值计算

4.13.1 基本概念

设函数 f 在某一有界区域 Ω 内有定义并且是连续的，计算积分

$$I(f)=\iint\limits_{\Omega} f(x,y)\mathrm{d}x\mathrm{d}y. \tag{4.13.1}$$

假定积分区域 Ω 是由两条连续单值的曲线 $y=\varphi(x)$，$y=\psi(x)$（$\varphi(x)\leqslant\psi(x)$），$a\leqslant x\leqslant b$ 和两条垂直线 $x=a$，$x=b$ 围成的(见图 4.4).

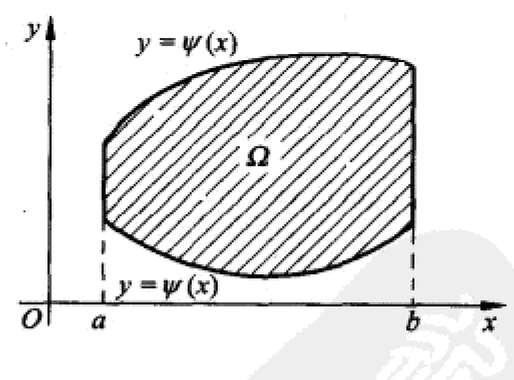


图 4.4

可以把积分(4.13.1)表示成

$$\iint\limits_{\Omega} f(x,y)\mathrm{d}x\mathrm{d}y=\int_a^b \mathrm{d}x\int_{\varphi(x)}^{\psi(x)} f(x,y)\mathrm{d}y.$$

从而有

$$\int_a^{\infty} e^{-x} f(x) dx = e^{-a} \sum_{k=1}^{n} A_k f(x_k + a) + \frac{(n!)^2}{(2n)!} f^{(2n)}(\xi), \tag{4.12.3}$$

其中 $A_k, x_k(k=1,2,\cdots,n)$ 分别为拉盖尔公式中的求积系数和求积节点,见附表 4.21.

(2) 广义拉盖尔公式

权函数 $\rho(x)=x^{\alpha}e^{-x}(\alpha>-1)$,

$$\int_0^{\infty} x^{\alpha} e^{-x} f(x) dx = \sum_{k=1}^{n} A_k f(x_k) + \frac{n!\Gamma(n+\alpha+1)}{(2n)!} f^{(2n)}(\xi)$$

$$(0<\xi<\infty), \tag{4.12.4}$$

其中节点 x_k 是广义拉盖尔多项式

$$L_n^{(\alpha)}(x) = \sum_{m=0}^{n} (-1)^m \binom{n+\alpha}{n-m} \frac{1}{m!} x^m \tag{4.12.5}$$

的零点,$A_k=n!\ \Gamma(n+\alpha+1)x_k/[L_{n+1}^{(\alpha)}(x_k)]^2$,

$$\Gamma(\beta) = \int_0^{\infty} e^{-t} t^{\beta-1} dt.$$

广义拉盖尔多项式可以由递推关系

$$(n+1)L_{n+1}^{(\alpha)}(x) = [(2n+\alpha+1)-x]L_n^{(\alpha)}(x) - (n+\alpha)L_{n-1}^{(\alpha)}(x) \tag{4.12.6}$$

得到,其中

$$L_0^{(\alpha)}(x) = 1,\quad L_1^{(\alpha)}(x) = 1+\alpha-x.$$

4.12.4 极限过程

$$\int_0^{\infty} f(x) dx = \lim_{r\to\infty} \int_0^{r} f(x) dx$$

提供了极限过程. 令 $0<r_0<r_1<\cdots$ 是趋向于∞的数列. 记

$$\int_0^{\infty} f(x) dx = \int_0^{r_0} f(x) dx + \int_{r_0}^{r_1} f(x) dx + \int_{r_1}^{r_2} f(x) dx + \cdots$$

设

$$F(x)=\int_{\varphi(x)}^{\psi(x)}f(x,y)\mathrm{d}y,$$

则有

$$\iint_{\Omega}f(x,y)\mathrm{d}x\mathrm{d}y=\int_{a}^{b}F(x)\mathrm{d}x.$$

上式右边的定积分可以采用数值方法求积.

$$\iint_{\Omega}f(x,y)\mathrm{d}x\mathrm{d}y=\sum_{k=1}^{n}C_kF(x_k) \tag{4.13.2}$$

其中 $x_k\in[a,b]$,$C_k(k=1,2,\cdots,n)$为求积系数. 对于

$$F(x_k)=\int_{\varphi(x_k)}^{\psi(x_k)}f(x_k,y)\mathrm{d}y,$$

也可以用求积公式

$$F(x_k)=\sum_{l=1}^{m_k}B_{kl}f(x_k,y_l) \tag{4.13.3}$$

来求得,其中 B_{kl} 是求积系数.

由式(4.13.2)和式(4.13.3)可以得到

$$\iint_{\Omega}f(x,y)\mathrm{d}x\mathrm{d}y=\sum_{k=1}^{n}\sum_{l=1}^{m_k}C_kB_{kl}f(x_k,y_l), \tag{4.13.4}$$

此公式称为**乘积型求积公式**.

4.13.2 梯形公式及其复合公式

设积分区域是矩形

$$R=\{(x,y)\mid a\leqslant x\leqslant A,\quad b\leqslant y\leqslant B\},$$

它的每一边平行于坐标轴. 令

$$x_0=a,\quad x_1=A,\quad y_0=b,\quad y_1=B$$

于是得到 4 个点$(x_k,y_l)(k,l=0,1)$. 如果 f 在 R 内连续,则有

$$\iint_{R}f(x,y)\mathrm{d}x\mathrm{d}y=\int_{a}^{A}\mathrm{d}x\int_{b}^{B}f(x,y)\mathrm{d}y. \tag{4.13.5}$$

利用梯形公式计算内部积分

$$\iint_R f(x,y)\mathrm{d}x\mathrm{d}y = \frac{B-b}{2}\int_a^A [f(x,y_0)+f(x,y_1)]\mathrm{d}x,$$

对上式右边再次应用梯形公式，可得

$$\iint_R f(x,y)\mathrm{d}x\mathrm{d}y = \frac{1}{4}(B-b)(A-a)[f(x_0,y_0)+f(x_1,y_0)+ f(x_0,y_1)+f(x_1,y_1)]. \tag{4.13.6}$$

此公式称作**梯形公式**.

为了提高精度，可以采用复合求积公式，即把求积区域 R 划分为一组矩形．而在每个矩形上应用梯形求积公式.

设把矩形 R 的边分别分为 n 等分和 m 等分，这样便把 R 分为边长为 h 和 k 的 mn 个小矩形．在每个小矩形上应用梯形公式得

$$\iint_R f(x,y)\mathrm{d}x\mathrm{d}y \approx \frac{hk}{4}\sum_{i=0}^{n-1}\sum_{j=0}^{m-1}[f(x_i,y_j)+ f(x_i,y_{j+1})+f(x_{i+1},y_j)+f(x_{i+1},y_{j+1})],$$

其中 $x_i=ih(i=0,1,\cdots,n)$，$y_j=jk(j=0,1,\cdots,m)$.

上式可以改写为

$$\iint_R f(x,y)\mathrm{d}x\mathrm{d}y \approx \frac{kh}{4}\sum_{i=0}^{n}\sum_{j=0}^{m}\lambda_{ij}f(x_i,y_j), \tag{4.13.7}$$

其中 λ_{ij} 是下面矩阵 $\boldsymbol{\Lambda}$ 的相应的元素，

$$\boldsymbol{\Lambda} = \begin{pmatrix} 1 & 2 & 2 & \cdots & 2 & 2 & 1 \\ 2 & 4 & 4 & \cdots & 4 & 4 & 2 \\ 2 & 4 & 4 & \cdots & 4 & 4 & 2 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 2 & 4 & 4 & \cdots & 4 & 4 & 2 \\ 2 & 4 & 4 & \cdots & 4 & 4 & 2 \\ 1 & 2 & 2 & \cdots & 2 & 2 & 1 \end{pmatrix}$$

公式(4.13.7)称为**复合梯形公式**.

4.13.3 辛普森求积公式及其复合公式

取积分区域为

$$R = \{(x,y) \mid a \leqslant x \leqslant A, b \leqslant y \leqslant B\},$$

分别用点

$$x_0 = a, \quad x_1 = a + h, \quad x_2 = a + 2h = A$$

和

$$y_0 = b, \quad y_1 = b + k, \quad y_2 = b + 2k = B$$

划分区间$[a,A]$和$[b,B]$，其中

$$h = \frac{1}{2}(A - a), \quad k = \frac{1}{2}(B - b).$$

这样得到 9 个点$(x_i, y_j)(i,j=0,1,2)$，点的分布见图 4.5.

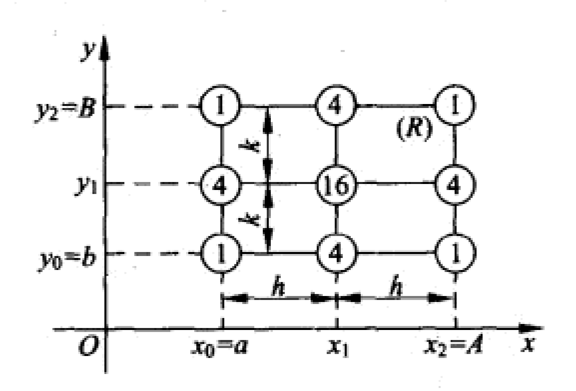


图 4.5

利用式(4.13.5)，并对内部积分用辛普森求积公式，有

$$\iint_R f(x,y)\mathrm{d}x\mathrm{d}y$$

$$= \frac{k}{3}\left[\int_a^A f(x,y_0)\mathrm{d}x + 4\int_a^A f(x,y_1)\mathrm{d}x + \int_a^A f(x,y_2)\mathrm{d}x\right]. \tag{4.13.8}$$

再对上式右边的每个积分应用辛普森求积公式，有

$$\iint_R f(x,y)\mathrm{d}x\mathrm{d}y \approx \frac{kh}{9}\{[f(x_0,y_0)+f(x_0,y_2)+f(x_2,y_0)+ f(x_2,y_2)]+4[f(x_1,y_0)+f(x_0,y_1)+ f(x_2,y_1)+f(x_1,y_2)]+16f(x_1,y_1)\}. \tag{4.13.9}$$

此公式称作**辛普森公式**. 如果令 σ_0 为被积函数 f 在矩形 R 的角点上的值之和, σ_1 为 f 在矩形 R 的每边中点上的值之和, σ_2 是 f 在矩形 R 的中心上的值, 那么公式(4.13.9)可以表示为

$$\iint_R f(x,y)\mathrm{d}x\mathrm{d}y = \frac{kh}{9}(\sigma_0+4\sigma_1+16\sigma_2).$$

上式右边的 $\sigma_i(i=0,1,2)$ 系数见图 4.5.

为提高求积精度, 一般采用复合公式. 设把矩形 R 的每边分别分成 n 等分和 m 等分, 这就得到了 nm 个小矩形. 再把每个小矩形等分为四部分, 这样就把 R 剖分成更小的矩形, 并把这些矩形的顶点用作求积公式中的节点.

令

$$h=\frac{A-a}{2n},\quad k=\frac{B-a}{2m},$$

那么节点的坐标为

$$\begin{cases} x_i = x_0 + ih & (x_0 = a, i = 0,1,\cdots,2n), \\ y_j = y_0 + jh & (y_0 = b, j = 0,1,\cdots,2m), \end{cases} \tag{4.13.10}$$

在第一次分 R 的 nm 个矩形上应用公式并记 $f_{ij}=f(x_i,y_j)$ 后, 有

$$\iint_R f(x,y)\mathrm{d}x\mathrm{d}y \approx \frac{hk}{9}\sum_{i=0}^{n-1}\sum_{j=0}^{m-1}[(f_{2i,2j}+f_{2i+2,j}+ f_{2i+2,2j+2}+f_{2i,2j+2})+4(f_{2i+1,2j}+f_{2i+2,2j+1}+ f_{2i+1,2j+2}+f_{2i,2j+1})+16f_{2i+1,2j+1}].$$

改写上式可以得到

$$\iint_R f(x,y)\mathrm{d}x\mathrm{d}y \approx \frac{hk}{9}\sum_{i=0}^{2n}\sum_{j=0}^{2m}\lambda_{ij}f_{ij}, \qquad (4.13.11)$$

其中系数 λ_{ij} 是矩阵 $\boldsymbol{\Lambda}$ 的相应的元素. $\boldsymbol{\Lambda}$ 定义为

$$\boldsymbol{\Lambda}=\begin{pmatrix} 1 & 4 & 2 & 4 & 2 & \cdots & 4 & 2 & 4 & 1 \\ 4 & 16 & 8 & 16 & 8 & \cdots & 16 & 8 & 16 & 4 \\ 2 & 8 & 4 & 8 & 4 & \cdots & 8 & 4 & 8 & 2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots \\ 2 & 8 & 4 & 8 & 4 & \cdots & 8 & 4 & 8 & 2 \\ 4 & 16 & 8 & 16 & 8 & \cdots & 16 & 8 & 16 & 8 \\ 1 & 4 & 2 & 4 & 2 & \cdots & 4 & 2 & 4 & 1 \end{pmatrix}.$$

例 4.13.1 用复合辛普森公式计算积分

$$\iint_R \ln(x+2y)\mathrm{d}x\mathrm{d}y \quad (\text{精确值为 } 0.4295545265),$$

其中 $R=\{(x,y)\mid 1.4\leqslant x\leqslant 2.0, 1.0\leqslant y\leqslant 1.5\}$.

解 先剖分 R，取 $h=0.15, k=0.25$，则有 $n=2, m=1$. 节点 $(x_i,y_j)\ i=0,1,2,3,4; j=0,1,2$ 分布见图 4.6.

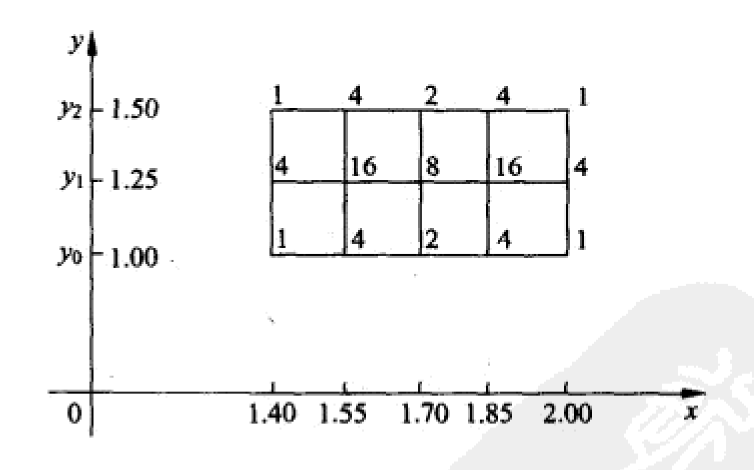


图 4.6

利用求积公式(4.13.11)，$f_{ij}=\ln(x_i+2y_j)$，系数 λ_{ij} 在图 4.6

上标出，则有

$$\iint_R \ln(x+2y)\,dy\,dx = \int_{1.4}^{2.0} dx \int_{1.0}^{1.5} \ln(x+2y)\,dy$$

$$\approx \frac{(0.15)(0.25)}{9} \sum_{i=0}^{4} \sum_{j=0}^{2} \lambda_{ij} \ln(x_j + 2y_j)$$

$$= 0.4295524387.$$

此计算值与精确值相比精确到 2.1×10^{-6}.

4.13.4 高斯型求积公式

在高斯-勒让德求积公式(4.5.7)

$$\int_{-1}^{1} g(t)\,dt \approx \sum_{i=1}^{n} A_i g(t_i)$$

对 $2n-1$ 次的代数多项式是精确成立的，上式中的节点 t_i 及系数 A_i 见表 4.23，定积分的高斯-勒让德求积公式很容易推广到重积分

$$I(f) = \int_{-1}^{1}\int_{-1}^{1} f(x,y)\,dx\,dy \tag{4.13.12}$$

的求积. 求积公式

$$\int_{-1}^{1}\int_{-1}^{1} f(x,y)\,dx\,dy \approx \sum_{i=1}^{n}\sum_{j=1}^{n} A_i A_j f(t_i, t_j) \tag{4.13.13}$$

对于二元函数 $f(x,y)=x^{\alpha}y^{\beta}$，$-1\leqslant x,y\leqslant 1$，$0\leqslant\alpha\leqslant 2n-1$，$0\leqslant\beta\leqslant 2n-1$ 精确成立，其中系数 A_i 及节点 t_i 仍由附表 4.21 确定. 由此可知，求积公式(4.13.13)对任意二元 $4n-2$ 次多项式精确成立. 求积公式(4.13.13)也称为重积分的高斯型求积公式.

例 4.13.2 用高斯型求积公式计算例 4.13.1 中的积分.

解 首先用线性变换

$$u = \frac{1}{2.0-1.4}(2x - 1.4 - 2.0),$$

$$v=\frac{1}{1.5-1.0}(2y-1.0-1.5),$$

将积分区域

$$R=\{(x,y)\mid 1.4\leqslant x\leqslant 2.0,1.0\leqslant y\leqslant 1.5\}$$

变换到正方形区域

$$\bar{R}=\{(x,y)\mid -1\leqslant u\leqslant 1,-1\leqslant v\leqslant 1\}.$$

经变换后积分为

$$\int_{1.4}^{2.0}\int_{1.0}^{1.5}\ln(x+2y)\mathrm{d}y\mathrm{d}x$$

$$=0.75\int_{-1}^{1}\int_{-1}^{1}\ln(0.3u+0.5v+4.2)\mathrm{d}v\mathrm{d}u$$

对上式右边积分,使用高斯求积公式(4.13.13),取 $n=3$,则采用节点

$$u_1=v_1=0,\quad u_0=v_0=-0.7745966692,$$

$$u_2=v_2=0.7745966692$$

相应的权

$$A_1=0.8888888889,\quad A_0=A_2=0.5555555556.$$

于是

$$\int_{1.4}^{2.0}\int_{1.0}^{1.5}\ln(x+2y)\mathrm{d}y\mathrm{d}x$$

$$\approx\sum_{i=0}^{2}\sum_{j=0}^{2}A_iA_j\ln(3u_i+5v_j+4.2)=0.4295545313.$$

此结果与精确值相比精确到 4.8×10^{-9}.

注意到此例仅计算 6 次函数值,而例 4.13.1 则计算了 15 次函数值,可知高斯求积公式是很实用的.

4.13.5 一般积分区域

考虑积分区域为一般的曲线围成的区域 Ω,则构造一个矩形 R 使 $R\supset\Omega$,且 R 的边平行于坐标轴(见图 4.7).

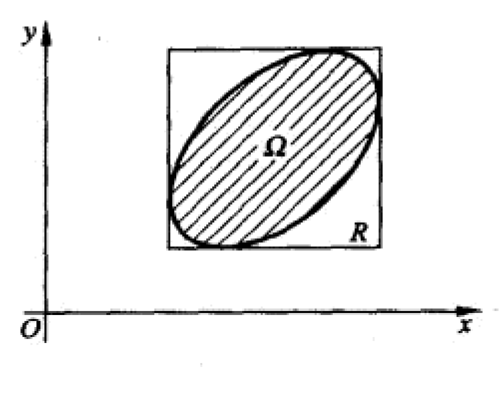


图 4.7

考虑辅助函数 f^*,其定义为

$$f^*(x,y)=\begin{cases}f(x,y), & (x,y)\in\Omega,\\ 0, & (x,y)\in R-\Omega,\end{cases}$$

显然有

$$\iint_{\Omega} f(x,y)\mathrm{d}x\mathrm{d}y=\iint_{R} f^*(x,y)\mathrm{d}x\mathrm{d}y.$$

上式右边的积分区域为矩形,因此可以采用已介绍的各种方法来求积.

4.14 数值微分的基本方法

4.14.1 数值微分的概念

在微分学中,函数的导数是通过极限定义的,但当函数用表格给出时,就不可用定义来求其导数,只能用近似方法求数值导数.最简单的数值微分公式是用差商近似地代替微商,常见的有

$$f'(x)\approx\frac{f(x+h)-f(x)}{h},$$

$$f'(x)\approx\frac{f(x)-f(x-h)}{h},$$

$$f'(x) \approx \frac{f(x+h)-f(x-h)}{2h}.$$

使用近似公式 $f'(x) \approx \dfrac{f(x+h)-f(x-h)}{2h}$ 的方法称为**中点方法**.

为了使用上述近似公式的方法来近似计算微商，首先必须选取合适的步长，为此要进行误差分析. 假定 f 是一个光滑的函数，利用泰勒展开有

$$f(x \pm h) = f(x) \pm h f'(x) + \frac{h^2}{2!} f''(x) \pm \frac{h^3}{3!} f''(x) + \frac{h^4}{4!} f^{(4)}(x) \pm \frac{h^5}{5!} f^{(5)}(x) + \cdots$$

如果用中点公式进行分析，则有

$$\frac{f(x+h)-f(x-h)}{2h} = f'(x) + \frac{h^2}{3!} f'''(x) + \frac{h^4}{5!} f^{(5)}(x) + \cdots$$

故步长越小，结果越准确. 但是，事实并非如此，因为当 h 很小时，由于 $f(x+a)$ 与 $f(x-a)$ 很接近，直接相减会造成有效数字的严重损失，所以必须适当选取 h.

例 4.14.1 用中点公式计算 $f(x)=\sqrt{x}$ 在 $x=2$ 处的一阶导数(导数精确值 $f'(2)=0.353553$).

解 计算采用公式

$$\frac{\sqrt{2+h}-\sqrt{2-h}}{2h} \approx f'(2),$$

取 4 位数字计算，结果见表 4.19.

表 4.19

h	$\frac{1}{2h}(\sqrt{2+h}-\sqrt{2-h})$	h	$\frac{1}{2h}(\sqrt{2+h}-\sqrt{2-h})$	h	$\frac{1}{2h}(\sqrt{2+h}-\sqrt{2-h})$
1	0.3660	0.05	0.3530	0.001	0.3500
0.5	0.3564	0.01	0.3500	0.0005	0.3000
0.1	0.3535	0.005	0.3500	0.0001	0.3000

可以看出，$h=0.1$ 的逼近效果最好，如果 h 再缩小，则逼近的效果反而差.

4.14.2 用插值多项式求数值微分

设 f 是定义在$[a,b]$上的函数，并在$[a,b]$上给定 $n+1$ 个节点 $x_0,x_1,\cdots,x_n$，则可以求得 f 在这些节点上的插值多项式 P_n. 若取拉格朗日形式，则

$$P_n(x)=\sum_{i=0}^{n}f(x_i)l_i(x),\quad x\in[a,b],$$

其中

$$l_i(x)=\prod_{\substack{j=0\\j\neq i}}^{n}\left(\frac{x-x_j}{x_i-x_j}\right)\quad(i=0,1,\cdots,n).$$

利用插值多项式 P_n，可以把 f 表示为

$$f=P_n+R_n,\tag{4.14.1}$$

其中 R_n 为插值多项式 P_n 的余项. 如果 f 有 $n+1$ 阶的连续导数，则拉格朗日形式的插值多项式的余项为

$$R_n(x)=\frac{f^{(n+1)}(\xi)}{(n+1)!}\prod_{j=0}^{n}(x-x_j),\quad \xi\in(a,b).$$

此余项也可以写成均差形式，即

$$R_n(x)=f[x_0,x_1,\cdots,x_n,x]\prod_{j=0}^{n}(x-x_j).$$

如果 $f\in C^{n+2}[a,b]$，那么可以得到

$$f'(x)=P'_n(x)+f[x_0,x_1,\cdots,x_n,x]\frac{\mathrm{d}}{\mathrm{d}x}\prod_{j=0}^{n}(x-x_j)+$$

$$f[x_0,x_1,\cdots,x_n,x,x]\prod_{j=0}^{n}(x-x_j),$$

从而有

$$f'(x)=P'_n(x)+\frac{f^{(n+1)}(\xi)}{(n+1)!}\frac{\mathrm{d}}{\mathrm{d}x}\prod_{j=0}^{n}(x-x_j)+$$

$$\frac{f^{(n+2)}(\eta)}{(n+2)!}\prod_{j=0}^{n}(x-x_j),$$

其中 $\xi,\eta\in(a,b)$.

特别地,在某一节点 x_k 上的导数为

$$f'(x_k)=P'_n(x_k)+\frac{1}{(n+1)!}f^{(n+1)}(\xi(x_k))\prod_{\substack{j=0\\j\neq k}}^{n}(x_k-x_j).$$

(4.14.2)

此公式称为 $f'(x_k)(k=0,1,\cdots,n)$的$(n+1)$点公式.

在等距节点的情况下,设间距为 h,令 $x=x_0+th$,利用牛顿前插公式有

$$P_n(x)=y_0+t\Delta y_0+\frac{t(t-1)}{2!}\Delta^2 y_0+\cdots+\frac{t(t-1)\cdots(t-n+1)}{n!}\Delta^n y_0,$$

其中 $y_0=f(x_0)$. 于是得到

$$f'(x_0)\approx P'_n(x_0)=\frac{1}{h}\left[\Delta y_0-\frac{1}{2}\Delta^2 y_0+\frac{1}{3}\Delta^3 y_0+\cdots+\frac{(-1)^{n-1}}{n}\Delta^n y_0\right]$$

(4.14.3)

此式适用于表头. 同样,适用于表末($t\leqslant 0$)的公式是

$$f'(x_0)\approx\frac{1}{h}\left(\Delta y_{-1}+\frac{1}{2}\Delta^2 y_{-2}+\frac{1}{3}\Delta^3 y_{-3}+\cdots+\frac{1}{n}\Delta^n y_{-n}\right).\quad(4.14.4)$$

对于表中间,则采用斯特林(Stirling)插值多项式的微商,有

$$f'(x_0)\approx\frac{1}{h}\left(\frac{\Delta y_{-1}+\Delta y_0}{2}-\frac{1}{6}\frac{\Delta^3 y_{-2}+\Delta^3 y_{-1}}{2}+\frac{1}{30}\frac{\Delta^5 y_{-3}+\Delta^5 y_{-2}}{2}+\cdots\right).\quad(4.14.5)$$

在实际计算中,使用公式(4.14.5)更方便,并且比公式(4.14.3)和公式(4.14.4)有更高的精度.

如果在公式(4.14.3)和公式(4.14.4)中仅取一项,则有

$$\begin{cases} f'(x_0) \approx \dfrac{1}{h}(y_1 - y_0), \\ f'(x_0) \approx \dfrac{1}{h}(y_0 - y_{-1}). \end{cases} \tag{4.14.6}$$

此式称为**二点公式**.

如果在公式(4.14.3)和公式(4.14.4)中取前面两项,而在公式(4.14.5)中取一项,则得到**三点公式**

$$\begin{cases} f'(x_0) \approx \dfrac{1}{2h}(-3y_0 + 4y_1 - y_2), \\ f'(x_0) \approx \dfrac{1}{2h}(y_{-2} - 4y_{-1} + 3y_0), \\ f'(x_0) \approx \dfrac{1}{2h}(y_1 - y_{-1}). \end{cases} \tag{4.14.7}$$

此外,常用的还有**五点公式**

$$\begin{cases} f'(x_0) \approx \dfrac{1}{12h}(-25y_0 + 48y_1 - 36y_2 + 16y_3 - 3y_4), \\ f'(x_0) \approx \dfrac{1}{12h}(-3y_{-1} - 10y_0 + 18y_1 - 16y_2 + y_3), \\ f'(x_0) \approx \dfrac{1}{12h}(y_{-2} - 8y_{-1} + 8y_1 - y_2), \\ f'(x_0) \approx \dfrac{1}{12h}(-y_{-3} + 6y_{-2} - 18y_{-1} + 10y_0 + 3y_1), \\ f'(x_0) \approx \dfrac{1}{12h}(3y_{-4} - 16y_{-3} + 36y_{-2} - 48y_{-1} + 25y_0). \end{cases} \tag{4.14.8}$$

对于给定的数据表,用五点公式(4.14.8)求节点上的导数值一般可以获得好的结果.

例 4.14.2 根据 $f(x)=\sqrt{x}$ 的数值(见表 4.20),利用五点公式求出节点上的近似导数值.

解 结果见表 4.20.

表 4.20

x_i	$f(x_i)$	$f'(x_i)$	计算值
100	10.000000	0.050000	0.050000
101	10.049875	0.049752	0.049751
102	10.099504	0.049507	0.049507
103	10.148891	0.049266	0.049267
104	10.198039	0.049029	0.049029
105	10.246950	0.048795	0.048795

用插值多项式 P_n 来代替函数 f 将产生截断误差，公式(4.14.3)和公式(4.14.4)可以用 $\frac{1}{n+1}h^n|f^{(n+1)}(\xi)|$ 来估计，其中 ξ 分别属于区间 (x_0, x_n) 和 (x_{-n}, x_0). 通常，$f^{(n+1)}$ 并不知道，所以这样的截断误差估计很难应用. 此外，由于舍入误差的影响，在数值微分的计算中一般不采用高阶差商，而对于仅包含低阶差商的数值微分公式，截断误差是容易估计的. 例如，对于公式(4.14.7)的第三式，如果三阶差商的变化是充分光滑的，那么可以近似地用
$\frac{1}{6h}\frac{|\Delta^3 y_{-2}+\Delta^3 y_{-1}|}{2}$ 来估计.

利用插值多项式求数值导数时，插入多项式 P_n 可收敛到 f，但 P'_n 不一定收敛到 f'；此外，当 h 缩小时，截断误差减小，但舍入误差可能增加，因此计算必须注意.

4.14.3 将微分问题化为积分问题

微分是积分的逆运算，因此可借助于数值积分来计算数值微分.

设 f 是一个充分光滑的函数，其导数为 φ. 由积分定义有

$$f(x) = f(\hat{x}) + \int_{\hat{x}}^{x} \varphi(t)\mathrm{d}t, \tag{4.14.9}$$

其中 $\hat{x}$ 为任意指定的数. 设 $x_i = x_0 + ih (i=0,1,\cdots,n)$ 为一组等距节点,并设 $y_k = f(x_k)$. 在公式(4.14.9)中取 $\hat{x} = x_{k-1}, x = x_{k+1}$,于是式(4.14.9)变为

$$f(x_{k+1}) = f(x_{k-1}) + \int_{x_{k-1}}^{x_{k+1}} \varphi(t)\,\mathrm{d}t \quad (k = 1,2,\cdots,n-1). \tag{4.14.10}$$

对上式右端的积分采用不同的求积公式就得到不同的数值微分公式.

(1) 对积分采用中点公式

$$\int_{x_{k-1}}^{x_{k+1}} \varphi(t)\,\mathrm{d}t = 2h\varphi(x_k) + \frac{(2h)^3}{24}\varphi''(\xi_k),$$

从而得到**中点微分公式**

$$f'(x_k) = \varphi(x_k) = \frac{f(x_{k+1}) - f(x_{k-1})}{2h} - \frac{h^2}{6} f'''(\xi_k), \tag{4.14.11}$$

其中 $x_{k-1} \leqslant \xi_k \leqslant x_{k+1} (k=1,2,\cdots,n-1)$. 可以看出,此式与公式(4.14.7)的第三式是一样的.

(2) 如果对式(4.14.10)中的积分采用辛普森求积公式,则有

$$\int_{x_{k-1}}^{x_{k+1}} \varphi(t)\,\mathrm{d}t = \frac{h}{3}[\varphi(x_{k-1}) + 4\varphi(x_k) + \varphi(x_{k+1})] - \frac{h^5}{90} f^{(5)}(\xi_k), \tag{4.14.12}$$

其中 $x_{k-1} < \xi_k < x_k$.

如果记 φ_k 为 $f'(x_k)$ 的近似值,且在上式中略去高阶项,那么从式(4.14.10)可得到**辛普森数值微分公式**

$$\varphi_{k-1} + 4\varphi_k + \varphi_{k+1} = \frac{3(y_{k+1} - y_{k-1})}{h} \quad (k = 1,2,\cdots,n-1). \tag{4.14.13}$$

式(4.13.13)有 $n+1$ 个未知函数 $\varphi_0, \varphi_1, \cdots, \varphi_n$,但只有 $n-1$ 个方程. 如果在端点有 $\varphi(x_0) = f'(x_0), \varphi(x_n) = f'(x_n)$,那么

式(4.14.13)可以写作

$$\begin{pmatrix} 4 & 1 & & & & \\ 1 & 4 & 1 & & & \\ & 1 & 4 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & 4 & 1 \end{pmatrix} \begin{pmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \\ \vdots \\ \varphi_{n-1} \end{pmatrix} = \begin{pmatrix} 3(y_2 - y_0)/h & -f'(x_0) \\ 3(y_3 - y_1)/h & \\ 3(y_4 - y_2)/h & \\ \vdots & \\ 3(y_n - y_{n-1})/h & -f'(x_n) \end{pmatrix}, \tag{4.14.14}$$

于是,数值微分化为解线性代数方程组的问题.显然,用追赶法容易求解.

例 4.14.3 给定 $f(x)=\sqrt{x}$在 $x=100,101,102,103,104,105$ 的一个表,求函数 f 在 $x=101,102,103,104$ 上的一阶导数值.

解 假定 f 在 $x=100$ 及 $x=105$ 处的一阶导数值已知,按方程组(4.14.14)解之,结果见表 4.21.

表 4.21

x	$\sqrt{x}$	$(\sqrt{x})'$	导数近似值
100	10.000000	0.05000000	
101	10.049875	0.049751859	0.049751859
102	10.099504	0.049507377	0.049507376
103	10.148891	0.049266463	0.049266463
104	10.198039	0.049029033	0.049029033
105	10.246950	0.049795003	

从算例看出,采用辛普森数值微分公式计算的精度相当高.

如果端点的导数值并不知道,对 $\varphi(x_1)$ 和 $\varphi(x_n)$ 的近似值 φ_1 和 φ_n 则可用中点微分公式来近似,这样辛普森微分公式(4.14.13)可以写作

$$
\begin{cases}
\varphi_1 = \dfrac{y_2 - y_0}{2h}, \\
\varphi_{k-1} + 4\varphi_k + \varphi_{k+1} = \dfrac{3(y_{k+1} - y_{k-1})}{h}, \quad k = 1,2,\cdots,n-1 \\
\varphi_{n-1} = \dfrac{y_n - y_{n-2}}{2h},
\end{cases}
\tag{4.14.15}
$$

当 $n=3$ 时，有

$$
\begin{cases}
\varphi_1 = \dfrac{y_2 - y_0}{2h}, \\
\varphi_0 + 4\varphi_1 + \varphi_2 = \dfrac{3(y_2 - y_0)}{h}, \\
\varphi_1 + 4\varphi_2 + \varphi_3 = \dfrac{3(y_3 - y_1)}{h}, \\
\varphi_2 = \dfrac{y_3 - y_1}{2h},
\end{cases}
$$

求解得

$$
\begin{cases}
f'(x_0) \approx \dfrac{1}{2h}[2(y_2 - y_0) - (y_3 - y_1)], \\
f'(x_1) \approx \dfrac{y_2 - y_0}{2h}, \\
f'(x_2) \approx \dfrac{y_3 - y_1}{2h}, \\
f'(x_3) \approx \dfrac{1}{2h}[2(y_3 - y_1) - (y_2 - y_0)].
\end{cases}
\tag{4.14.16}
$$

4.14.4 用三次样条函数求数值微分

设在区间$[a,b]$上给定一个剖分

$$
a = x_0 < x_1 < \cdots < x_{n-1} < x_n = b,
$$
$$
x_k = a + kh, \quad h = \frac{1}{n}(b-a)
$$

及相应于$\{x_k\}$的函数值$\{y_k\}$($k=0,1,\cdots,n$)，如果给定适当的边界条件，则可以惟一确定三次样条函数 S，其表达式为

$$
S(x) = \sum_{j=-1}^{n+1} c_j \Omega_3\left(\frac{x - x_j}{h}\right), \quad x \in [a,b].
$$

对上式两边求导数，则有

$$f'(x) \approx S'(x) = \frac{1}{h}\sum_{j=-1}^{n+1} c_j \Omega'_3\left(\frac{x-x_j}{h}\right), \quad (4.14.17)$$

令 $x=x_k$，则有

$$f'(x_k) \approx S'(x_k) = \frac{1}{h}\sum_{j=-1}^{n+1} c_j \Omega'_3\left(\frac{x_k-x_j}{h}\right)$$

$$= \frac{1}{h}\sum_{j=-1}^{n+1} c_j \Omega'_3(k-j),$$

由此得

$$f'(x_k) \approx \frac{1}{2h}(c_{k+1} - c_{k-1}) \quad (k=1,2,\cdots,n-1). \tag{4.14.18}$$

此公式说明，函数 $f(x)$ 在 x_k 处的导数值近似地由它的三次样条函数的系数 c_{k+1}, c_{k-1} 线性表示.

如果 f 有一阶连续导数，那么 S' 在 $[a,b]$ 上一致收敛到 f'，因此当 h 取得充分小时用样条求数值微分是很精确的.

4.14.5 二阶导数

如果 f 在区间 $[x_0-h, x_0+h]$ 上有四阶连续的导数，则利用泰勒级数展开有

$$f''(x_0) = \frac{1}{h^2}[f(x_0-h) - 2f(x_0) + f(x_0+h)] - \frac{h^2}{12}f^{(4)}(\xi),$$

其中 $x_0-h<\xi<x_0+h$. 如果略去上式右边的最后一项，可得数值微分公式

$$f''(x_0) \approx \frac{1}{h^2}[f(x_0-h) - 2f(x_0) + f(x_0+h)].$$

利用插值多项式来求二阶数值微分是经常使用的方法. 设 P_n 是 f 的插值多项式，对于任意 x，有

$$f''(x) \approx P''_n(x).$$

对于等距节点，有以下常用公式，为简便起见，记 $y_k=f(x_k)$.

$$
\begin{cases}
f''(x_0) \approx \dfrac{1}{h^2}(y_0 - 2y_1 + y_2), \\
f''(x_0) \approx \dfrac{1}{h^2}(y_{-1} - 2y_0 + y_1), \\
f''(x_0) \approx \dfrac{1}{h^2}(y_{-2} - 2y_{-1} + y_0);
\end{cases}
\tag{4.14.19}
$$

$$
\begin{cases}
f''(x_0) \approx \dfrac{1}{6h^2}(12y_0 - 30y_1 + 24y_2 - 6y_3), \\
f''(x_0) \approx \dfrac{1}{6h^2}(6y_{-1} - 12y_0 + 6y_1), \\
f''(x_0) \approx \dfrac{1}{6h^2}(-6y_{-3} + 24y_{-2} - 30y_{-1} + 12y_0);
\end{cases}
\tag{4.14.20}
$$

$$
\begin{cases}
f''(x_0) \approx \dfrac{1}{12h^2}(35y_0 - 104y_1 + 114y_2 - 56y_3 + 11y_4), \\
f''(x_0) \approx \dfrac{1}{12h^2}(11y_{-1} - 20y_0 + 6y_1 + 4y_2 - y_3), \\
f''(x_0) \approx \dfrac{1}{12h^2}(-y_{-2} + 16y_{-1} - 30y_0 + 16y_1 - y_2), \\
f''(x_0) \approx \dfrac{1}{12h^2}(-y_{-3} + 4y_{-2} - 6y_{-1} + 20y_0 + 11y_1), \\
f''(x_0) \approx \dfrac{1}{12h^2}(11y_{-4} - 56y_{-3} + 114y_{-2} - 104y_{-1} + 35y_0).
\end{cases}
\tag{4.14.21}
$$

二阶数值微分也可以用三次样条函数方法来计算. 设在$[a,b]$上给定一个剖分

$$
a = x_0 < x_1 < \cdots < x_n = b,
$$

$$
x_i = a + ih, \quad h = \frac{b-a}{n}
$$

及相应于这个剖分$\{x_i\}$的函数值$\{y_i\}$. 如果再给定适当的边界条件就可以构造出惟一的三次样条函数 S,其表达式为

$$
S(x) = \sum_{j=-1}^{n+1} c_j \Omega_3\left(\frac{x - x_j}{h}\right).
$$

对式(4.14.17)再求一次导数有

$$f''(x) \approx S''(x) = \frac{1}{h^2}\sum_{j=-1}^{n+1} c_j \Omega''_3\left(\frac{x-x_j}{h}\right).$$

令 $x=x_i$,得

$$f''(x_i) \approx S''(x_i) = \frac{1}{h^2}(c_{i+1} - 2c_i + c_{i-1}).$$

利用样条函数的 c_i 和 y_i 之间的关系式

$$\frac{1}{6}c_{i-1} + \frac{2}{3}c_i + \frac{1}{6}c_{i+1} = y_i$$

就得到

$$f''(x_i) \approx S''(x_i) = \frac{6}{h^2}(y_i - c_i) \quad (i = 1,2,\cdots,n-1). \tag{4.14.22}$$

有些问题还可以应用下面的方法来提高精度.令

$$\tilde{y}_i = S'(x_i) \quad (i = 0,1,\cdots,n),$$
$$\tilde{y}'_0 = f''(x_0) \quad \tilde{y}'_n = f''(x_n),$$

满足这个条件的插值函数记为 $\tilde{S}'$:

$$\tilde{S}'(x) = \sum_{j=-1}^{n+1} \tilde{c}_j \Omega_3\left(\frac{x-x_j}{h}\right).$$

对 $\tilde{S}'$求导,有

$$\tilde{S}''(x) = \frac{1}{h}\sum_{j=-1}^{n+1} \tilde{c}_j \Omega'_3\left(\frac{x-x_j}{h}\right).$$

如果以 $\tilde{S}''$来近似 f'',那么对 $x=x_i$ 就有

$$f''(x_i) = \frac{1}{2h}(\tilde{c}_{i+1} - \tilde{c}_{i-1}). \tag{4.14.23}$$

4.15 数值微分的外推算法

在数值积分中应用外推算法产生了龙贝格求积方法,这是非常有效的.同样,在数值微分中采用外推算法也是很有效的.

用中心差分来计算导数值有

$$f'(x) \approx G(h) = \frac{f\left(x+\frac{h}{2}\right)-f\left(x-\frac{h}{2}\right)}{h}. \quad (4.15.1)$$

利用泰勒级数展开有

$$f'(x) - G(h) = a_1 h^2 + a_2 h^4 + a_3 h^6 + \cdots \quad (4.15.2)$$

其中常数 a_i 与 h 无关. 利用理查森外推算法, 取 $q=\frac{1}{2}$有

$$\begin{cases} G_1(h) = G(h), \\ G_{m+1}(h) = \dfrac{G_m\left(\frac{h}{2}\right)-\left(\frac{1}{2}\right)^{2m} G_m(h)}{1-\left(\frac{1}{2}\right)^{2m}}, \quad m = 1,2,\cdots. \end{cases} \quad (4.15.3)$$

这个算法见表 4.22.

表 4.22 ①,…,⑩表示计算步骤

①$G(h)$				
②$G\left(\frac{h}{2}\right)$	③$G_2(h)$			
④$G\left(\frac{h}{2^2}\right)$	⑤$G_2\left(\frac{h}{2}\right)$	⑥$G_3(h)$		
⑦$G\left(\frac{h}{2^3}\right)$	⑧$G_2\left(\frac{h}{2^2}\right)$	⑨$G_3\left(\frac{h}{2}\right)$	⑩$G_4(h)$	
⋮	⋮	⋮	⋮	…

利用这个算法的误差为

$$f'(x) - G_{m+1}(h) = a_{m+1}^{(m+1)} h^{2(m+1)} + \cdots$$

所以, 当 m 越大时越精确, 但受舍入误差的影响, m 不能取得很大.

例 4.15.1 用外推算法计算 $f(x)=\tan^{-1}x$ 在 $x=\sqrt{2}$处的导数.

解 计算步骤及结果如下:

$G(1) = 0.3926991$,

$G\left(\frac{1}{2}\right) = 0.3487710$, $G_2(1) = 0.3341283$,

$G\left(\frac{1}{4}\right) = 0.3371938$, $G_2\left(\frac{1}{2}\right) = 0.3333348$, $G_3(1) = 0.3332819$,

$G\left(\frac{1}{8}\right) = 0.3342981$, $G_2\left(\frac{1}{4}\right) = 0.3333329$,

$G_3\left(\frac{1}{2}\right) = 0.333328$, $G_4(1) = 0.3333336$,

$G\left(\frac{1}{16}\right) = 0.3335748$, $G_2\left(\frac{1}{8}\right) = 0.3333336$, $G_3\left(\frac{1}{4}\right) = 0.3333337$,

$G_4\left(\frac{1}{2}\right) = 0.3333337$, $G_5(1) = 0.3333337$.

可以看出，由 $G(1), G\left(\frac{1}{2}\right), \cdots, G\left(\frac{1}{16}\right)$ 经简单运算便得 $f'(x)\Big|_{x=\sqrt{2}} = \frac{1}{1+x^2}\Big|_{x=\sqrt{2}} = \frac{1}{3}$ 的相当精确的近似值.

外推算法也可应用于计算二阶数值微商. 仍采用中心差分公式来计算导数值

$$f''(x) \approx S(h) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2},$$

利用 $f(x+h), f(x-h)$ 在 x 处展成泰勒级数，经运算可得

$$f''(x) - S(h) = c_1 h^2 + c_2 h^4 + c_3 h^6 + \cdots$$

由此可以导出如同式(4.15.3)的算法.

4.16 附表

4.16.1 高斯-勒让德求积公式的节点和系数

求积公式为

$$\int_{-1}^{1} f(x)\mathrm{d}x \approx \sum_{i=1}^{n} A_i f(x_i),$$

节点$=\pm x_i$（勒让德多项式的零点），系数$=A_i$，节点数$=n$，见表4.23.

表 4.23

n	x_i	A_i
2	0.57735 02691 89626	1.00000 00000 00000
3	0.00000 00000 00000	0.88888 88888 88889
	0.77495 66692 41483	0.55555 55555 55556
4	0.33998 10435 84856	0.65214 51548 62546
	0.86113 63115 94053	0.34785 48451 37454
5	0.00000 00000 00000	0.56888 88888 88889
	0.53846 93101 05683	0.47862 86704 99366
	0.90617 98459 38664	0.23692 68850 56189
6	0.23861 91860 83197	0.46791 39345 72691
	0.66120 93864 66265	0.36076 15730 48139
	0.93246 95142 03152	0.17132 44923 79170
7	0.00000 00000 00000	0.41795 91836 73496
	0.40584 51513 77397	0.38183 00505 05119
	0.74153 11855 99394	0.27970 53914 89277
	0.94910 79123 42759	0.12948 49661 68870
8	0.18343 46424 95650	0.36268 37833 78362
	0.52553 24099 16329	0.31370 66458 77887
	0.79666 64774 13627	0.22238 10344 53374
	0.96028 98564 97536	0.10122 85362 90376
9	0.00000 00000 00000	0.33023 93550 01260
	0.32425 34234 03809	0.31234 70770 40003
	0.61337 14327 00590	0.26061 06964 02935
	0.83603 11073 26636	0.18064 81606 94857
	0.96816 02395 07626	0.08127 43883 61574

续表

n	x_i	A_i
10	0.14887 43389 81631	0.29552 42247 14753
	0.43339 53941 29247	0.26926 67193 09996
	0.67940 95682 99024	0.21908 63625 15982
	0.86506 33666 88985	0.14945 13491 50581
	0.97390 65285 17172	0.06667 13443 08688
12	0.12523 34085 11469	0.24914 70458 13403
	0.36783 14989 98180	0.23349 25365 38355
	0.58731 79542 86617	0.20316 74267 23066
	0.76990 26741 94305	0.16007 83285 43346
	0.90411 72563 70475	0.10693 93259 95318
	0.98156 06342 46719	0.04717 53363 86512
16	0.09501 25098 37637 440185	0.18945 06104 55068 496285
	0.28160 35507 79258 913230	0.18260 34150 44923 588867
	0.45801 67776 57227 386342	0.16915 65193 95002 538189
	0.61787 62444 02643 748447	0.14959 59888 16576 732081
	0.75540 44083 55003 033895	0.12462 89712 55533 872052
	0.86563 12023 87831 743880	0.09515 85116 82492 784810
	0.94457 50230 73232 576078	0.06225 35239 38647 892863
	0.98940 09349 91649 932596	0.02715 24594 11754 094852
20	0.07652 65211 33497 333755	0.15275 33871 30725 850698
	0.22778 58511 41645 078080	0.14917 29864 72603 746788
	0.37370 60887 15419 560673	0.14209 61093 18382 051329
	0.51086 70019 50827 098004	0.13168 86384 49176 626898
	0.63605 36807 26515 025453	0.11819 45319 61518 417312
	0.74633 19064 60150 792614	0.10193 01198 17240 435037
	0.83911 69718 22218 823395	0.08327 67415 76704 748725
	0.91223 44282 51325 905868	0.06267 20483 34109 063570
	0.96397 19272 77913 791268	0.04060 14298 00386 941331
	0.99312 85991 85094 924786	0.01761 40071 39152 118312

4.16.2 高斯-拉盖尔求积公式的节点和系数

求积公式为

$$\int_0^{\infty} e^{-x} f(x) dx \approx \sum_{i=1}^{n} A_i f(x_i),$$

$$\int_0^{\infty} g(x) dx \approx \sum_{i=1}^{n} A_i e^{x_i} g(x_i),$$

节点 $= x_i$（拉盖尔多项式的零点），系数 $= A_i$，节点数 $= n$，见表 4.24.

表 4.24

n	x_i	A_i	$A_i e^{x_i}$
2	0.58578 64376 27	0.853553 390593	1.53332 603312
	3.41421 35623 73	0.146446 609407	4.45095 733505
3	0.41577 45567 83	0.711093 009929	1.07769 285927
	2.29428 03602 79	0.278517 733569	2.76214 296190
	6.28994 50829 37	0.103892 565016 $\times 10^{-1}$	5.60109 462543
4	0.32254 76896 19	0.603154 104342	0.83273 91238 38
	1.74576 11011 58	0.357418 692438	2.04810 243845
	4.53662 02969 21	0.388879 085150 $\times 10^{-1}$	3.63114 630582
	9.39507 09123 01	0.539294 705561 $\times 10^{-3}$	6.48714 508441
5	0.26356 03197 18	0.521755 610583	0.67909 40422 08
	1.41340 30591 07	0.398666 811083	1.63848 787360
	3.59642 57710 41	0.759424 496817 $\times 10^{-1}$	2.76944 324237
	7.08581 00058 59	0.361175 867992 $\times 10^{-2}$	4.31565 690092
	12.64080 08442 76	0.233699 723858 $\times 10^{-4}$	7.21918 635435
6	0.22284 66041 79	0.458964 673950	0.57353 55074 23
	1.18893 21016 73	0.417000 830772	1.36925 259071
	2.99273 63260 59	0.113373 382074	2.26068 459338
	5.77514 35691 05	0.103991 974531 $\times 10^{-1}$	3.35052 458236
	9.83746 74183 83	0.261017 202815 $\times 10^{-3}$	4.88682 680021
	15.98287 39806 02	0.898547 906430 $\times 10^{-6}$	7.84901 594560

续表

n	x_i	A_i	$A_i e^{x_j}$
10	0.13779 34705 40	0.308441 115765	0.35400 97386 07
	0.72945 45495 03	0.401119 929115	0.83190 23010 44
	1.80834 29017 40	0.218068 287612	1.33028 856175
	3.40143 36978 55	0.620874 560987 $\times 10^{-1}$	1.86306 390311
	5.55249 61400 64	0.950151 697518 $\times 10^{-2}$	2.45025 555808
	8.33015 27467 64	0.753008 388588 $\times 10^{-3}$	3.12276 415514
	11.84378 58379 00	0.282592 334960 $\times 10^{-4}$	3.93415 269556
	16.27925 78313 78	0.424931 398496 $\times 10^{-6}$	4.99241 487219
	21.99658 58119 81	0.183956 482398 $\times 10^{-8}$	6.57220 248513
	29.92069 70122 74	0.991182 721961 $\times 10^{-11}$	9.78469 584037

4.16.3 高斯-埃尔米特求积公式的节点和系数

求积公式为

$$\int_{-\infty}^{\infty} e^{-x^2} f(x) dx \approx \sum_{i=1}^{n} A_i f(x_i),$$

$$\int_{-\infty}^{\infty} g(x) dx = \sum_{i=1}^{n} A_i e^{x_i^2} g(x_i),$$

节点$=\pm x_i$(埃尔米特多项式的零点),系数$=A_i$,节点数$=n$,见表 4.25.

表 4.25

n	x_i	A_i	$A_i e^{x_i^2}$
2	0.70710 67811 86548	0.886226 92545 28	1.46114 11826 611
3	0.00000 00000 00000	1.18163 59006 04	1.18163 59006 037
	1.22474 48713 91589	0.295408 97515 09	1.32393 11752 136
4	0.52464 76232 75290	0.804914 09000 55	1.05996 44828 950
	1.65068 01238 85785	0.813128 35447 25 $\times 10^{-1}$	1.24022 58176 958

续表

n	x_i			A_i			$A_i e^{x_i^2}$		
5	0.00000	00000	00000	0.945308	72048	29	0.94530	87204	829
	0.95857	24646	13819	0.393619	32315	22	0.98658	09967	514
	2.02018	28704	56086	0.199532	42059	05×10^{-1}	1.18148	86255	360
6	0.43607	74119	27617	0.724629	59522	44	0.87640	13344	362
	1.33584	90740	13697	0.157067	32032	29	0.93558	05576	312
	2.35060	49736	74492	0.453000	99055	09×10^{-2}	1.13690	83326	745
7	0.00000	00000	00000	0.810264	61755	68	0.81026	46175	568
	0.81628	78828	58965	0.425607	25201	01	0.82868	73032	836
	1.67355	16287	67471	0.545155	82819	13×10^{-1}	0.89718	46002	252
	2.65196	13568	35233	0.971781	24509	95×10^{-3}	1.10133	07296	103
8	0.38118	69902	07322	0.661147	01255	82	0.76454	41286	517
	1.15719	37124	46780	0.207802	32581	49	0.79289	00483	864
	1.98165	67566	95843	0.170779	83007	41×10^{-1}	0.86675	26065	634
	2.93063	74202	57244	0.199604	07221	14×10^{-3}	1.07193	01442	480
9	0.00000	00000	00000	0.720235	21560	61	0.72023	52156	061
	0.72355	10187	52838	0.432651	55900	26	0.73030	24527	451
	1.46855	32892	16668	0.884745	27394	38×10^{-1}	0.76460	81250	946
	2.26658	05845	31843	0.494362	42755	37×10^{-2}	0.84175	27014	787
	3.19099	32017	81528	0.396069	77263	26×10^{-4}	1.04700	35809	767
10	0.34290	13272	23705	0.610862	63373	53	0.68708	18539	513
	1.03661	08287	89514	0.240138	61108	23	0.70329	63231	049
	1.75668	36492	99882	0.338743	94455	48×10^{-1}	0.74144	19319	436
	2.53273	16742	32790	0.134364	57467	81×10^{-2}	0.82066	61264	048
	3.43615	91188	37738	0.764043	28552	33×10^{-5}	1.02545	16913	657
12	0.31424	03762	54359	0.570135	23626	25	0.62930	78743	695
	0.94778	83912	40164	0.260492	31026	42	0.63962	12320	203
	1.59768	26351	52605	0.516079	85615	88×10^{-1}	0.66266	27732	669
	2.27950	70805	01060	0.390539	05846	29×10^{-2}	0.70522	03661	122
	3.02063	70251	20890	0.857368	70435	88×10^{-4}	0.78664	39394	633
	3.88972	48978	69782	0.265855	16843	56×10^{-6}	0.98969	90470	923

5 方程求根

5.1 方程求根与二分法

5.1.1 方程求根与根的隔离

在科学技术的数学问题中，经常遇到求解函数方程

$$f(x)=0, \tag{5.1.1}$$

这里 $f(\cdot)$是单变量 x 的函数，它可以是代数多项式，即

$$f(x)=a_0x^n+a_1x^{n-1}+\cdots+a_{n-1}x+a_n \quad (a_0\neq 0), \tag{5.1.2}$$

也可以是超越函数. 满足 $f(x^*)=0$ 的值 x^* 就是方程(5.1.1)的解，x^* 称作方程的根，又称作函数 $f(\cdot)$的**零点**. 如果 $f(x)$可分解为

$$f(x)=(x-\alpha)^m g(x), \tag{5.1.3}$$

m 为正整数且 $g(\alpha)\neq 0$，则称 α 为 $f(x)=0$ 的 **m 重根**. $m=1$ 称为**单根**，$m>1$ 称 α 为方程(5.1.1)的 m 重根，或 α 为 $f(x)$的 m 重零点，若 α 是 $f(x)$的 m 重零点，且 $g(x)$充分光滑，则

$$f(\alpha)=f'(\alpha)=\cdots=f^{(m-1)}(\alpha)=0, \quad f^{(m)}(\alpha)\neq 0.$$

方程求根首先要解决根是否存在的问题，如果是由式(5.1.2)表示的代数多项式，则由代数基本定理可知，在复数域内方程有 n 个根(含复根，m 重根为 m 个根)，对一般函数方程，若 $f(x)$在区间$[a,b]$上连续，且 $f(a)f(b)<0$，则方程(5.1.1)在$[a,b]$上至少有一个实根，$[a,b]$称为有根区间.

在根存在的前提下，求方程的根通常是采用逐次逼近思想构

造的迭代方法,这类方法产生一个序列 $x_0,x_1,\cdots$,使它收敛于方程的根 α,为此需要先找到有根区间$[a,b]$或近似根 α 的初始值 x_0,这就是根的隔离问题.通常可用逐次搜索法求有根区间$[a,b]$,具体做法可从 $x_0=a$ 出发取步长 $h=\frac{b-a}{n}$(n 为正整数),令 $x_k=a+kh(k=0,1,\cdots,n)$,从左至右检查 $f(x_k)$的符号,如发现节点 x_k 与端点 a 的函数值异号,则得到一个缩小的有根区间$[x_{k-1},x_k]$,其宽度为 h.再检查下去,只要发现相邻两点函数值异号则可得一个缩小的有根区间.

例 5.1.1 给定方程

$$f(x)=x^3-x-1=0,$$

由于 $f(0)<0,f(2)>0$,故$[0,2]$是有根区间.若取 $h=0.5$,从左向右检查 $f(x_k)$的符号(见下表),可发现$[1,1.5]$是一个缩小的有根区间.

x_k	0	0.5	1.0	1.5	2.0
$f(x_k)$	$-$	$-$	$-$	$+$	$+$

用这种逐步搜索的方法进行实根隔离的关键是选取步长 h,如 h 选得太大,则有的根可能被遗漏;如 h 选得太小,则可得到较精确的有根区间,但工作量较大.要选择适当的 h,使之既能把根隔离开来,工作量又不太大,可采用二分法(bisection method),它可看作逐步搜索法的改进.

5.1.2 二分法

设 $f(\cdot)$在$[a,b]$连续,假定 $f(a)<0,f(b)>0$,取中点 $x_0=\frac{a+b}{2}$,检查 $f(x_0)$符号.若 $f(x_0)=0$,则 x_0 就是一个根;若 $f(x_0)>0$,记 a 为 a_1,x_0 为 b_1,则得有根区间$[a_1,b_1]$;若 $f(x_0)<0$,记 x_0 为

a_1，b 为 b_1，则得有根区间$[a_1,b_1]$. 后两种情况都得到有根区间$[a_1,b_1]$，它的长度为原区间的一半. 对$[a_1,b_1]$，令 $x_1=\frac{a_1+b_1}{2}$，再施以同样方法，可得新的有根区间$[a_2,b_2]$，它的长度为$[a_1,b_1]$的一半，如此反复进行下去可得到一系列有根区间

$$[a,b]\supset[a_1,b_1]\supset\cdots\supset[a_n,b_n]\supset\cdots$$

其中每一个区间都是前一区间的一半，见图 5.1.

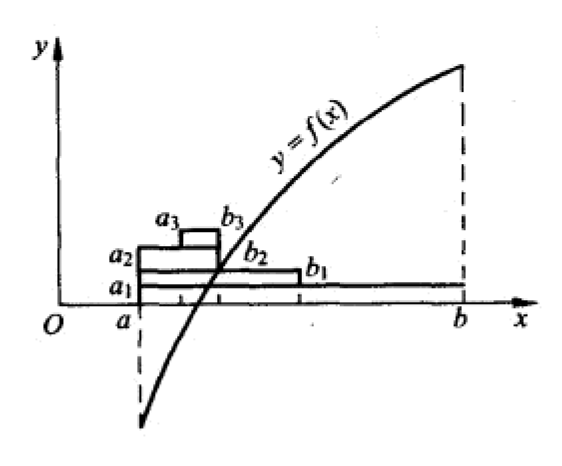


图 5.1

因此，$[a_n,b_n]$的长度为

$$b_n-a_n=\frac{b-a}{2^n},$$

当 $n\to\infty$时趋于零，且$\lim\limits_{n\to\infty}x_n=\lim\limits_{n\to\infty}\frac{a_n+b_n}{2}=x^*$，这就是方程的根. 而 $x_n=\frac{a_n+b_n}{2}$即为方程的近似根，且有误差估计

$$|x_n-x^*|\leqslant\frac{b-a}{2^{n+1}}. \tag{5.1.4}$$

例 5.1.2 用二分法求

$$f(x)=x^3-x-1=0$$

在区间$[1,1.5]$的一个实根，准确到小数点后第 2 位数.

解 这里 $a=1,b=1.5$，根据上述步骤取区间中点 $x_0=1.25$，检查 $f(x_0)$的符号，决定新区间. 如此反复得到一系列区间如下：

$f(1)<0$	有根区间
$f(1.5)>0$	$[1,1.5]$
$f(1.25)<0$	$[1.25,1.5]$
$f(1.375)>0$	$[1.25,1.375]$
$f(1.3125)<0$	$[1.3125,1.375]$
$f(1.34375)>0$	$[1.3125,1.34375]$
$f(1.3281)>0$	$[1.3125,1.3281]$
$f(1.3203)<0$	$[1.3203,1.3281]$

取 $x_6=1.3242$，误差限 $|x_6-x^*|<\frac{0.5}{2^7}<0.005$，故 x_6 即为所求近似根，实际上根 $x^*=1.324717\cdots$.

上述二分法的优点是计算简单，收敛性有保证，缺点是收敛不够快，特别是精度要求高时工作量大，而且，不能求复根及双重根.

5.2 迭代法及其收敛性

5.2.1 不动点迭代法

为了构造方程求根的迭代公式，通常可将方程(5.1.1)改写成等价形式

$$x=g(x), \tag{5.2.1}$$

则求 x^* 满足 $f(x^*)=0$ 等价于求 x^* 使 $x^*=g(x^*)$，称 x^* 为 $g(x)$的**不动点**. 于是求 $f(x)$的零点等价于求 $g(x)$的不动点. 若已知方程(5.1.1)的一个近似根 x_0，代入式(5.2.1)右端，即可求得 $x_1=g(x_0)$，如此反复迭代，可得到迭代序列

$$x_{k+1}=g(x_k) \quad (k=0,1,\cdots), \tag{5.2.2}$$

$g(x)$称为**迭代函数**. 如果对初始近似 x_0，迭代序列$\{x_k\}$有极限

$$\lim_{k\to\infty} x_k = x^*,$$

则称迭代过程(5.2.2)**收敛**,且对式(5.2.2)取极限得到 $x^* = g(x^*)$. x^* 就是 $g(x)$ 的不动点,故方法(5.2.2)称为不动点迭代法. x^* 就是方程(5.2.1)的根.

方程(5.2.1)的求根问题,从几何图像考察就是在 xOy 平面上确定曲线 $y=x$ 与 $y=g(x)$ 的交点 P^*. 用迭代法(5.2.2)求根就是从 $y=x_0$ 与 $y=g(x)$ 的交点出发逐次求点 P^* 的横坐标 x^*,图 5.2(a)表示迭代序列(5.2.2)收敛,(b)表示迭代不收敛.

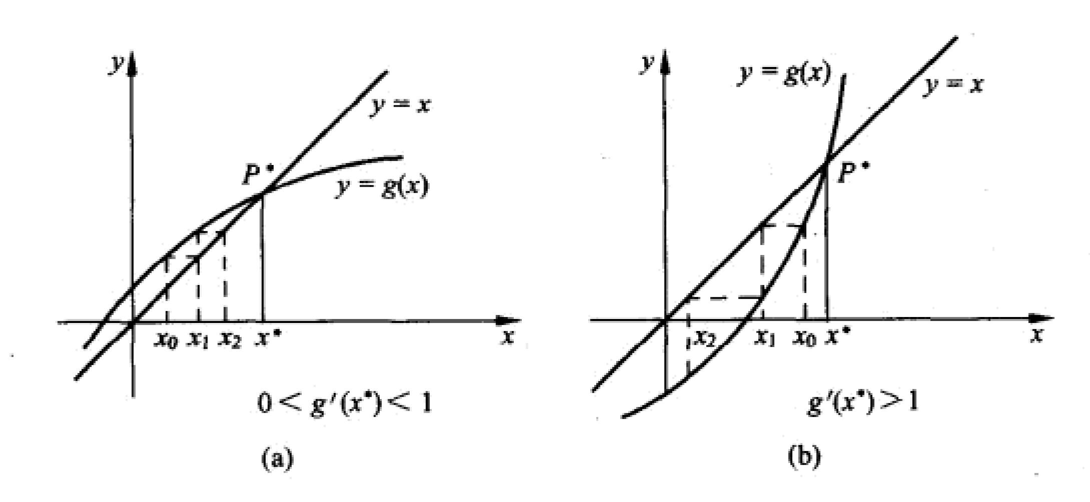


图 5.2

例 5.2.1 用迭代法求方程

$$f(x) = x^3 - x - 1 = 0$$

在 $x_0=1.5$ 附近的根 x^*.

解 将方程改写成

$$x = \sqrt[3]{1+x},$$

并建立迭代程序

$$x_{k+1} = \sqrt[3]{1+x_k} \quad (k = 0,1,\cdots). \tag{5.2.3}$$

只要按式(5.2.3)逐步计算 $g(x_k)=\sqrt[3]{1+x_k}$,便可得到迭代结果:

$x_0=1.5$, $x_1=1.35721$, $x_2=1.33086$,

$x_3=1.32588$, $x_4=1.32494$, $x_5=1.32476$,

$x_6=1.32473$, $x_7=1.32472$, $x_8=1.32472$.

这说明迭代序列(5.2.3)收敛,且 $x_8=1.32472$ 为方程的近似根.但如果将方程改写为 $x=x^3-1$,并建立迭代公式

$$x_{k+1}=x_k^3-1, \tag{5.2.4}$$

则 $x_0=1.5, x_1=2.375, x_2=12.39, \cdots$. x_k 的值越算越大,说明迭代序列(5.2.4)不收敛,因此这个迭代序列不能用.

例 5.2.1 表明原方程化为方程(5.2.1)的形式不同,得到的迭代法有的收敛,有的发散,只有收敛的迭代法(5.2.2)才有意义.为此需要研究 $g(x)$ 不动点存在性与迭代法(5.2.2)的收敛性.

5.2.2 不动点存在性与迭代法收敛性

若方程(5.2.1)在 $[a,b]$ 上有根 x^*,则由迭代(5.2.2)得

$$x_{k+1}-x^*=g(x_k)-g(x^*)=g'(\xi)(x_k-x^*), \tag{5.2.5}$$

ξ 在 x_k 与 x^* 之间,当然 $\xi\in[a,b]$. 因此,当 $|g'(x)|\leqslant L<1$ 时,由式(5.2.5)可得

$$|x_{k+1}-x^*|\leqslant L|x_k-x^*|\leqslant\cdots\leqslant L^{k+1}|x_0-x^*|.$$

当 $k\to\infty$ 时,$|x_k-x^*|\to0$,迭代序列 $\{x_k\}$ 收敛到根 x^*. 实际上,对迭代序列(5.2.2)有以下不动点存在和收敛定理.

定理 5.2.2 假定 $g(x)$ 在 $[a,b]$ 上可导,$\max\limits_{a\leqslant x\leqslant b}|g'(x)|\leqslant L<1$,且当 $x\in[a,b]$ 时 $g(x)\in[a,b]$,则 $g(x)$ 在 $[a,b]$ 上存在惟一的不动点 x^*,且对 $\forall x_0\in[a,b]$,由迭代法(5.2.2)生成的序列 $\{x_k\}$ 收敛于 x^*,并有误差估计

$$|x_k-x^*|\leqslant\frac{L^k}{1-L}|x_1-x_0|. \tag{5.2.6}$$

在例 5.2.1 中，当 $g(x)=\sqrt[3]{1+x}$ 时 $g'(x)=\frac{1}{3}(1+x)^{-\frac{2}{3}}$，在区间[1,2]中，$\max\limits_{1\leqslant x\leqslant 2}|g'(x)|=\frac{1}{3}\frac{1}{\sqrt[3]{4}}<0.21<1$ 且当 $x\in[1,2]$ 时，$1\leqslant g(x)\leqslant 2$，故迭代序列(5.2.3)收敛. 对迭代公式(5.2.4)，$g'(x)=3x^2$，当 $x\in[1,2]$ 时，$g'(x)\geqslant 3$，定理条件不满足，因此迭代公式(5.2.4)不能使用. 定理 5.2.2 既给出了迭代法(5.2.2)收敛到根 x^* 的充分条件，也给出了方程(5.2.1)在$[a,b]$上根的存在性，它是一个大范围收敛的定理，条件较强，不易满足，通常只在所求根 x^* 附近研究序列$\{x_k\}$的收敛性及收敛速度.

定义 5.2.3 对任何 $x_0\in R$，$R=\{x:|x-x^*|\leqslant\delta,\delta>0\}$，迭代法(5.2.2)生成的序列$\{x_k\}$均收敛到 x^*，则称此迭代序列具有**局部收敛性**(local convergence).

定理 5.2.4 假定 x^* 是方程(5.2.1)的根，$g'(x)$在 x^* 的邻域连续，且$|g'(x^*)|<1$，则迭代法(5.2.2)是局部收敛的.

迭代序列$\{x_k\}$的局部收敛性反映了迭代序列在 $g(x)$不动点 x^* 附近的收敛情况，必须知道 $g(x)$的不动点 x^* 才能使用，或在 x^* 的邻域$|x-x^*|<\delta$ 有$|g'(x)|\leqslant L<1$，而$|g'(x^*)|$的大小与迭代序列$\{x_k\}$的收敛快慢有关，先看下例.

例 5.2.5 用不同方法求方程 $x^2-3=0$ 的根 $x^*=\sqrt{3}$.

解 本例 $f(x)=x^2-3$，可改写为不同的等价形式 $x=g(x)$，g 的不动点 $x^*=\sqrt{3}$. 下面给出 4 种不同的迭代法：

(1) $x_{k+1}=x_k^2+x_k-3$，　$g(x)=x^2+x-3$，

$$g'(x)=2x+1,\quad g'(x^*)=g'(\sqrt{3})=2\sqrt{3}+1>1.$$

(2) $x_{k+1}=\frac{3}{x_k}$，　$g(x)=\frac{3}{x}$，

$$g'(x)=-\frac{3}{x^2},\quad g'(x^*)=-1.$$

(3) $x_{k+1}=x_k-\frac{1}{4}(x_k^2-3)$，　$g(x)=x-\frac{1}{4}(x^2-3)$，

$$g'(x)=1-\frac{1}{2}x,\quad g'(x^*)=1-\frac{\sqrt{3}}{2}\approx 0.134<1.$$

(4) $x_{k+1}=\frac{1}{2}\left(x_k+\frac{3}{x_k}\right)$，　$g(x)=\frac{1}{2}\left(x+\frac{3}{x}\right)$，

$$g'(x)=\frac{1}{2}\left(1-\frac{3}{x^2}\right),\quad g'(x^*)=g'(\sqrt{3})=0.$$

若取 $x_0=2$，对上述 4 种迭代法各计算 3 步结果列于表 5.1.

表 5.1　计算$\sqrt{3}$的不同迭代

k	x_k	迭代法(1)	迭代法(2)	迭代法(3)	迭代法(4)
0	x_0	2	2	2	2
1	x_1	3	1.5	1.75	1.75
2	x_2	9	2	1.73475	1.732143
3	x_3	87	1.5	1.732361	1.732051
⋮	⋮	⋮	⋮	⋮	⋮

注意$\sqrt{3}=1.7320508\cdots$，从表 5.1 看到迭代法(1)及(2)均不收敛，它们均不满足定理 5.2.4 中局部收敛条件，而迭代法(3)和(4)均满足局部收敛条件，但迭代法(4)比(3)收敛更快，因迭代法(4)中 $g'(x^*)=0$. 为了衡量迭代法(5.2.2)收敛速度的快慢，有下面定义.

定义 5.2.6　迭代序列$\{x_k\}$收敛于 x^*，如果存在实数 $p\geqslant 1$ 和常数 $C\neq 0$，且当 $k\geqslant k_0$ 时 $x_k\neq x^*$，渐近关系式

$$\lim_{k\to\infty}\frac{x_{k+1}-x^*}{(x_k-x^*)^p}=C\neq 0$$

成立，则称迭代序列$\{x_k\}$是 **p 阶收敛的**(order p convergence). $p=1$称为**线性收敛**(linear convergence)，$p>1$ 称为**超线性收敛**(superlinear convergence)，$p=2$ 称为**平方收敛**(quadratic convergence).

若记

$$\overline{x}_{k+1} = x_k - \frac{(\Delta x_k)^2}{\Delta^2 x_k} \quad (k = 0,1,\cdots), \qquad (5.3.1)$$

其中 $\Delta x_k = x_{k+1} - x_k$ 是 x_k 点的一阶差分，$\Delta^2 x_k = x_{k+2} - 2x_{k+1} + x_k$ 是 x_k 点的二阶差分. 故式(5.3.1)称为艾特肯加速方法，也称 Δ^2 加速方法. 可以证明

$$\lim_{k\to\infty} \frac{\overline{x}_{k+1} - x^*}{x_k - x^*} = 0.$$

它表明序列$\{\overline{x}_k\}$的收敛速度比$\{x_k\}$快.

例 5.3.1 用艾特肯加速方法求方程

$$x = \mathrm{e}^{-x}$$

在 $x_0 = 0.5$ 附近的根.

解 若用迭代法 $x_{k+1} = \mathrm{e}^{-x_k}$ 求根，计算 18 步可得 $x_{18} = 0.56714$. 用艾特肯加速方法，则 $g(x_k) = \mathrm{e}^{-x_k}$，代入公式 $x_{k+1} = g(x_k)$，由式(5.3.1)计算一步得

$$x_0 = 0.5, \quad x_1 = 0.60653, \quad x_2 = 0.54524, \quad \overline{x}_1 = 0.56762.$$

计算第二步时应从 $\overline{x}_1$ 出发，于是可取 $x_1 = \overline{x}_1$，即

$$x_1 = 0.56762, \quad x_2 = 0.56687,$$
$$x_3 = 0.56730, \quad \overline{x}_2 = 0.56714,$$

这里只计算 2 步(相当于迭代 $x_{k+1} = g(x_k)$ 计算 4 步)就得到与不动点迭代法 x_{18} 的相同结果.

5.3.2 斯蒂芬森迭代法

艾特肯加速方法不管原序列$\{x_k\}$是怎样产生的，对$\{x_k\}$进行加速计算，得到序列$\{\overline{x}_k\}$. 若把这种加速技巧与不动点迭代法结合，则由式(5.3.1)可得如下迭代法：

$$\begin{cases} y_k = g(x_k), \quad z_k = g(y_k), \\ x_{k+1} = x_k - \dfrac{(y_k - x_k)^2}{z_k - 2y_k + x_k}, \end{cases} \qquad (5.3.2)$$

定理 5.2.7 设 x^* 是方程(5.2.1)的根，若 $g^{(p)}(x)$ 在根 x^* 附近连续，并且

$$g'(x^*)=\cdots=g^{(p-1)}(x^*)=0,\quad g^{(p)}(x^*)\neq 0,$$

则迭代法(5.2.2)是 p 阶收敛的.

上述定理表明，迭代法(5.2.2)收敛快慢取决于迭代函数 $g(x)$ 的选取，若 $\varphi'(x)\neq 0$，则该迭代法即使收敛也只能是线性收敛. 在例 5.2.5 中，迭代法(3)的 $g'(x^*)\neq 0$，它是线性收敛，而迭代法(4)中 $g'(x^*)=0$，而 $g''(x^*)=\dfrac{2}{\sqrt{3}}\neq 0$. 由定理 5.2.7 知 $p=2$，即该方法为二阶收敛，所以收敛很快.

对收敛慢或不收敛的迭代法，可用加速收敛的方法改善迭代的收敛性.

5.3 迭代法的加速收敛

5.3.1 艾特肯加速方法

如果迭代序列(5.2.2)收敛很慢，要达到要求的精度将使计算量很大，为此，需采用加速迭代收敛性的方法.

设 x_k 是根 x^* 的近似，$x_k-x^*\neq 0$，由式(5.2.5)可得

$$\frac{x_{k+1}-x^*}{x_k-x^*}=g'(\xi_k),\quad \xi_k \text{ 在 } x_k \text{ 与 } x^* \text{ 之间}.$$

假定 $g'(x)$ 在 x 变化时改变不大，可令 $g'(x)\approx L\neq 0$(线性)，于是可得

$$\frac{x_{k+1}-x^*}{x_k-x^*}\approx\frac{x_{k+2}-x^*}{x_{k+1}-x^*},$$

由此解出

$$x^*\approx\frac{x_{k+2}x_k-x_{k+1}^2}{x_{k+2}-2x_{k+1}+x_k}=x_k-\frac{(x_{k+1}-x_k)^2}{x_{k+2}-2x_{k+1}+x_k}.$$

$k=0,1,\cdots$，称为 Steffensen 迭代法，它是将原不动点迭代法(5.2.2)计算两次合并成一步得到，它可改写成为一种不动点迭代法：

$$x_{k+1}=\psi(x_k),\quad k=0,1,\cdots \tag{5.3.3}$$

其中

$$\psi(x)=x-\frac{[g(x)-x]^2}{g(g(x))-2g(x)+x}. \tag{5.3.4}$$

对不动点迭代法(5.3.3)有下面的局部收敛性定理.

定理 5.3.2 若 x^* 为式(5.3.4)定义的迭代函数 $\psi(x)$ 的不动点，则 x^* 也为 $g(x)$ 的不动点. 反之，若 x^* 为 $g(x)$ 的不动点，设 $g''(x)$ 连续，且 $g'(x^*)\neq 1$，则 x^* 是 $\psi(x)$ 的不动点，且迭代法(5.3.3)是二阶收敛的.

例 5.3.3 用 Steffensen 迭代法求解例 5.2.1 中方程的根，$g(x)$ 用不收敛的迭代公式(5.2.4).

解 式(5.2.4)中的 $g(x)=x^3-1$，现用迭代公式(5.3.2)计算，结果见表 5.2.

表 5.2

k	x_k	y_k	z_k
0	1.5	2.37500	12.3965
1	1.41629	1.84092	5.23888
2	1.35565	1.49140	2.31728
3	1.32895	1.34710	1.44435
4	1.32480	1.32518	1.32714
5	1.32472		

计算 5 步则得 6 位有效数字的近似根，它说明即使迭代法(5.2.4)不收敛，用 Steffensen 迭代法仍可能收敛，至于原来已收敛的迭代法(5.2.2)，则可达到二阶收敛.

5.4 牛顿法

5.4.1 牛顿法及其收敛性

牛顿法是通过非线性方程线性化得到迭代序列的一种方法.

对于方程

$$f(x)=0, \tag{5.4.1}$$

若已知根 x^* 的一个近似值 x_k,可将 $f(\cdot)$ 在 x_k 处展成一阶泰勒公式

$$f(x)=f(x_k)+f'(x_k)(x-x_k)+\frac{f''(\xi)}{2!}(x-x_k)^2.$$

取其线性部分近似,即用线性方程

$$f(x_k)+f'(x_k)(x-x_k)=0 \tag{5.4.2}$$

近似方程(5.4.1).若 $f'(x_k)\neq 0$,方程(5.4.2)的根记作 x_{k+1},则得

$$x_{k+1}=x_k-\frac{f(x_k)}{f'(x_k)} \quad (k=0,1,\cdots) \tag{5.4.3}$$

这就是牛顿法的迭代程序,它实际上是在点 x_k 处作曲线 $y=f(x)$ 的切线,并用切线与 x 轴交点 x_{k+1} 作为根 x^* 的新近似,如图 5.3 所示,故牛顿法也称**切线法**.

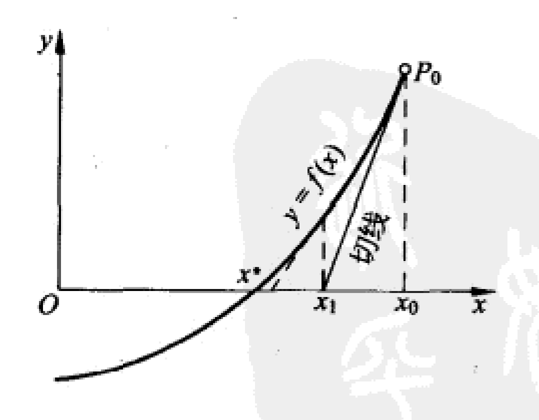


图 5.3

从迭代程序(5.4.3)可知，迭代函数为

$$g(x) = x - \frac{f(x)}{f'(x)},$$

于是

$$g'(x) = 1 - \frac{f'(x)}{f'(x)} + \frac{f(x)f''(x)}{[f'(x)]^2} = \frac{f(x)f''(x)}{[f'(x)]^2}.$$

若 $f(x^*)=0, f'(x^*)\neq 0$，则 $g'(x^*)=0$，且 $g''(x^*)\neq 0$. 由定理 5.2.7可知，牛顿法生成的迭代序列$\{x_k\}$在 x^* 附近平方收敛，说明牛顿法收敛很快.

例 5.4.1 用牛顿法求方程

$$f(x) = xe^x - 1 = 0$$

在 $x_0=0.5$ 附近的根.

解 此方程的牛顿迭代程序为

$$x_{k+1} = x_k - \frac{x_k - e^{-x_k}}{1 + x_k} \quad (k = 0,1,\cdots),$$

直接计算可得：

$x_0 = 0.5$， $x_1 = 0.57102$， $x_2 = 0.56716$， $x_3 = 0.56714$，

只算 3 步就达到 10^{-5}精度，可见牛顿法收敛很快.

将牛顿法应用于 $f(x)=x^2-c=0$，求得 $\sqrt{c}$ 的牛顿程序为

$$\begin{aligned} x_{k+1} &= x_k - \frac{x_k^2 - c}{2x_k} \\ &= \frac{1}{2}\left[x_k + \frac{c}{x_k}\right] \quad (k = 0,1,\cdots). \end{aligned} \tag{5.4.4}$$

如求$\sqrt{115}$，即 $c=115$，取 $x_0=10$，用公式(5.4.4)计算 3 步则得 $x_3=10.723805$，精度达到 10^{-6}. 每步只用一次除法和一次加法，计算量小，且对任意 $x_0>0$，迭代法(5.4.4)总收敛到$\sqrt{c}$，且是二阶收敛，因此式(5.4.4)可作为计算$\sqrt{x}$的标准子程序.

5.4.2 简化牛顿法与牛顿下山法

牛顿法的优点是收敛快且可用于求复根，缺点是每步要计算 $f(x_k)$ 及 $f'(x_k)$，计算量较大，且方法只是局部收敛. 若初始近似 x_0 给的不合适则不能保证其收敛性，为克服这两个缺点，可用下列方法.

(1) **简化牛顿法**，也称**平行弦法**，其迭代公式为

$$x_{k+1} = x_k - cf(x_k) \quad (k = 0,1,\cdots \quad c \neq 0), \tag{5.4.5}$$

迭代函数 $g(x)=x-cf(x), g'(x)=1-cf'(x)$,

若 $|g'(x^*)|<1$，则方法局部收敛，因此，只要 $0<c<\frac{2}{f'(x^*)}$ 时，$|g'(x^*)|=|1-cf'(x^*)|<1$，方法局部收敛. 在式(5.4.5)中取 $c=\frac{1}{f'(x_0)}$ 则称为简化牛顿法，这类方法每步只算一次 f 的值，计算量小，但方法只是线性收敛. 其几何意义是用 x_k 处的平行弦与 x 轴交点作为根 x^* 的近似.

(2) 牛顿下山法

牛顿法只具有局部收敛性，即初始近似 x_0 要在根 x^* 附近. 为了扩大收敛范围，可以采用牛顿下山程序

$$x_{k+1} = x_k - \lambda_k \frac{f(x_k)}{f'(x_k)} \quad (k = 0,1,\cdots), \tag{5.4.6}$$

其中参数 $0<\lambda_k\leqslant 1$，应满足下山条件：

$$|f(x_{k+1})|<|f(x_k)|, \tag{5.4.7}$$

故 λ_k 称为**下山因子**. 若令

$$\bar{x}_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)},$$

则牛顿下山法(5.4.6)等价于

$$x_{k+1} = \lambda_k \bar{x}_{k+1} + (1-\lambda_k)x_k. \tag{5.4.8}$$

使用牛顿下山法求根时，下山因子 λ_k 可用逐步搜索法确定，即先

令 $\lambda_k=1$，判断条件(5.4.7)是否成立，若不成立再将 λ_k 缩小$\frac{1}{2}$，直到条件(5.4.7)成立为止.

例 5.4.2 用牛顿法求方程

$$f(x)=x^3-x-1=0.$$

解 计算程序

$$x_{k+1}=x_k-\frac{x_k^3-x_k-1}{3x_k^2-1}\quad(k=0,1,\cdots).\qquad(5.4.9)$$

当 $x_0=1.5$ 时，计算 3 步得 $x_3=1.32472$，因为 x_0 与 x^* 很靠近，故收敛很快. 但如取 $x_0=0.6$，则由程序(5.4.9)求得 $x_1=17.9$，再算下去显然不会收敛. 如用牛顿下山法(5.4.8)，令 $\bar{x}_1=17.9$，从 $\lambda_0=1$开始逐次搜索，当 $\lambda_0=\frac{1}{32}$时，由牛顿下山法(5.4.8)可得

$$x_1=\frac{1}{32}\bar{x}_1+\frac{31}{32}x_0=1.140625,$$

满足条件 $|f(x_1)|<|f(x_0)|$，x_1 已修正了 $\bar{x}_1$ 的严重偏差，以后计算由于 $\lambda_k=1$ 就能使条件(5.4.7)成立. 因此牛顿下山法与牛顿法结果一样，$x_2=\bar{x}_2=1.366814$，$x_3=\bar{x}_3=1.32628$，$x_4=\bar{x}_4=1.32472$.

5.5 弦截法与抛物线法

牛顿法是用切线近似曲线 $y=f(x)$求得新近似值，它要计算导数 $f'(x_k)$. 当计算 $f'(x)$较困难时，往往用 $f(x)$在一些点上的函数值来近似，如用曲线上两点确定的直线来近似曲线求得方程的近似根. 这种方法称为**弦截法**. 如用曲线上三个点作抛物线近似曲线，求得方程新的近似根的方法称为**抛物线法**. 更一般的就是用 $y=f(x)$的插值多项式 $P_n(x)$近似 $f(x)$求方程的根，这就是插值求根法.

5.5.1 弦截法

如图 5.4,设曲线 $y=f(x)$ 上两点 $(x_0,f(x_0))$,$(x_1,f(x_1))$ 已知,通过这两点的直线方程为

$$P_1(x)=f(x_1)+\frac{f(x_1)-f(x_0)}{x_1-x_0}(x-x_1). \quad (5.5.1)$$

若 $f(x_1)\neq f(x_0)$,则直线 $y=P_1(x)$ 与 x 轴 $(y=0)$ 的交点为 $(x_2,0)$,其中

$$x_2=x_1-\frac{x_1-x_0}{f(x_1)-f(x_0)}f(x_1).$$

把 x_2 作为新的近似,再由 $(x_2,f(x_2))$ 与 $(x_1,f(x_1))$ 两点定出 x_3,依此类推,若两点 $(x_k,f(x_k))$ 与 $(x_{k-1},f(x_{k-1}))$ 已知,在 $f(x_k)\neq f(x_{k-1})$ 时,有

$$x_{k+1}=x_k-\frac{x_k-x_{k-1}}{f(x_k)-f(x_{k-1})}f(x_k) \quad (k=1,2,\cdots), \quad (5.5.2)$$

这就是弦截法,也称割线法.它相当于牛顿法(5.4.3)中导数 $f'(x_k)$ 用 x_{k-1} 与 x_k 的差商来代替.

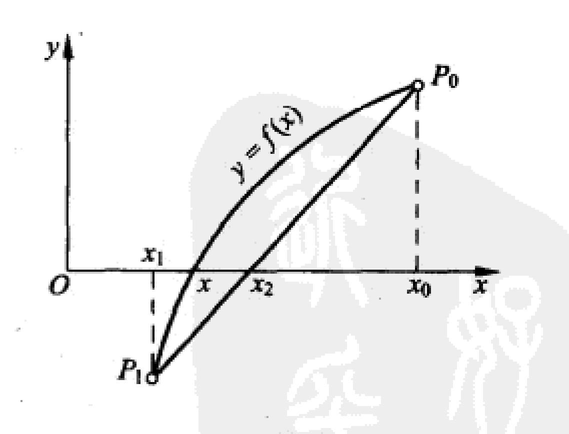


图 5.4

例 5.5.1 用弦截法求方程

$$f(x)=xe^x-1=0$$

在 $x_0=0.5$ 附近的根.

解 取 $x_0=0.5, x_1=0.6$,用弦截法公式(5.5.2)计算,每步算一次函数值,可求得

$$x_2=0.56532,\quad x_3=0.56709,\quad x_4=0.56714.$$

例 5.5.1 的结果与例 5.4.1 的牛顿法结果比较,可以看出弦截法的收敛速度也相当快.它有以下的局部收敛定理.

定理 5.5.2 假定 $f(\circ)$ 在根 x^* 的邻域 $\Delta=\{x:|x-x^*|<\delta\}$ 内具有二阶连续导数,且对任意 $x\in\Delta$ 有 $f'(x)\neq0$,初始近似 x_0, $x_1\in\Delta$,则当 Δ 充分小时,弦截法(5.5.2)生成的序列 $\{x_k\}$ 收敛到 x^*,且收敛阶数为 $p=\dfrac{1+\sqrt{5}}{2}\approx1.618$.

弦截法收敛阶数虽比牛顿法低,但它不用计算导数,且每步只算一次函数值,计算量少,因此,它是一个效率较高、适于在计算机上求方程根的方法.

5.5.2 试位法与斯蒂芬森方法

试位法是弦截法的一种变形,它选择 (x_n, f_n) 与 $(x_{n'}, f_{n'})$ 的割线代替通过 (x_n, f_n) 及 (x_{n-1}, f_{n-1}) 的割线,其中 $n'<n$,又使 $f_n\cdot f_{n'}<0$ 成立的最大下标 n',当然初始值 x_0 及 x_1 应使 $f_0\cdot f_1<0$,试位法的优点是当 $f(x)$ 连续则方法总是收敛的,但它一般只是线性收敛,当 $f(x)$ 在 $[x_0, x_1]$ 中为凸函数便是这种情况,如图 5.5 所示.因此它只用于求出好的初始近似,而不适用于根的精确化.

弦截法每步只算一个新的函数值,但它达不到二阶收敛.已经

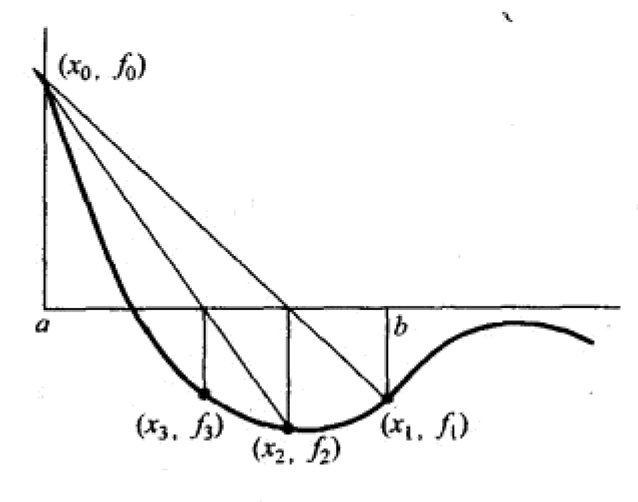


图 5.5

证明没有一种迭代法只算一次函数值就可达到二阶收敛，然而弦截法的一种变形

$$x_{n+1} = x_n - \frac{f(x_n)}{f(x_n + f(x_n)) - f(x_n)} f(x_n) \qquad (5.5.3)$$

是二阶收敛的，它要求计算两个函数值而不需要计算导数值. 公式(5.5.3)称为斯蒂芬森(Steffensen)方法，若令 $f(x)=g(x)-x$，则公式(5.5.3)就是在 5.3.2 节中得到的斯蒂芬森迭代公式(5.3.3)及式(5.3.4).

5.5.3 抛物线法

给定曲线 $y=f(x)$ 上三个不共线的点 $(x_0, f(x_0))$，$(x_1, f(x_1))$，$(x_2, f(x_2))$，通过这三点作抛物线 $y=P_2(x)$ 近似 $y=f(x)$，适当选取 $P_2(x)=0$ 的一个根记作 x_3，作为方程的新近似根. 这样确定的迭代过程就是**抛物线法**，也称**米勒(Müller)法**，几何图形就是用抛物线 $y=P_2(x)$ 与 x 轴的交点横坐标 x_3 作为根 x^* 的近似，如图 5.6 所示.

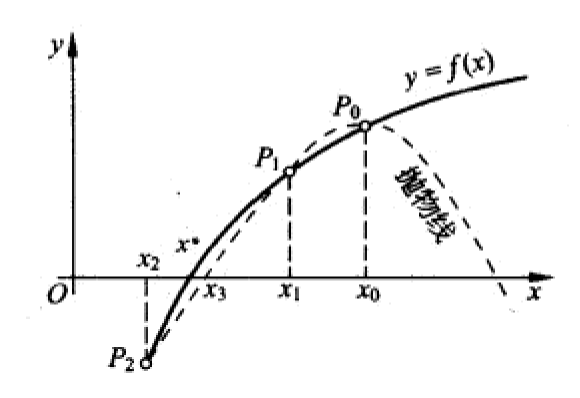


图 5.6

已知$(x_k, f(x_k))$,$(x_{k-1}, f(x_{k-1}))$,$(x_{k-2}, f(x_{k-2}))$三点不共线,由牛顿插值多项式有

$$\begin{aligned}P_2(x) &= f(x_k) + f(x_k, x_{k-1})(x - x_k) + \\ &\quad f(x_k, x_{k-1}, x_{k-2})(x - x_k)(x - x_{k-1}) \\ &= f(x_{k1}, x_{k-1}, x_{k-2})(x - x_k)^2 + (f(x_k, x_{k-1}) + \\ &\quad f(x_k, x_{k-1}, x_{k-2})(x_k - x_{k-1})) \cdot (x - x_k) + f(x_k),\end{aligned}$$

由于 $f(x_k, x_{k-1}, x_{k-2}) \neq 0$,故 $P_2(x) = 0$ 有两个根

$$x - x_k = \frac{-w \pm (w^2 - 4f(x_k)f(x_k, x_{k-1}, x_{k-2}))^{1/2}}{2f(x_k, x_{k-1}, x_{k-2})}, \tag{5.5.4}$$

其中

$$w = f(x_k, x_{k-1}) + f(x_k, x_{k-1}, x_{k-2})(x_k - x_{k-1}). \tag{5.5.5}$$

为避免有效位数损失,对式(5.5.4)采用有理化分子,并把 x 记作 x_{k+1},可得抛物线法计算公式

$$x_{k+1} = x_k - \frac{2f(x_k)}{w \pm (w^2 - 4f(x_k)f(x_k, x_{k-1}, x_{k-2}))^{1/2}} \quad (k = 2, 3, \cdots), \tag{5.5.6}$$

其中 w 由式(5.5.5)给出.在两根中应选与 x_k 靠近的值作为新近似根 x_{k+1},为此,取根式前的“±”号要与 w 同号.

例 5.5.3 用抛物线法求方程

$$f(x)=xe^{x}-1=0$$

在 $x_0=0.5$ 附近的根.

解 取 $x_0=0.5, x_1=0.6, x_2=0.56532$ 为初始近似,计算相应函数值与差商值:

$f(x_0)=-0.175639$, $f(x_1)=-0.093271$,

$f(x_2)=-0.005031$, $f(x_1,x_0)=2.68910$,

$f(x_2,x_1)=2.83373$, $f(x_2,x_1,x_0)=2.75694$,

代入式(5.5.5)算出 $w=2.75691$,再由式(5.5.6)求得 $x_3=0.56714$.此例精度已达到 10^{-5},说明抛物线法比弦截法收敛更快.事实上,在一定条件下可证明抛物线法的收敛阶数 $p\approx1.840$.抛物线法具有收敛快,能算复根等优点,缺点是要用开方运算,计算公式较复杂,因此在计算机上使用时,计算公式应做适当改变.其计算步骤如下:

1. 准备.选定初始近似 x_0,x_1,x_2,并计算相应的 f_0,f_1,f_2,及 $\lambda_2=\dfrac{x_2-x_1}{x_1-x_0}$.

2. 迭代.计算

$\delta_2=1+\lambda_2$,

$a=(\lambda_2 f_0-\delta_2 f_1+f_2)\lambda_2$,

$b=\lambda_2^2 f_0-\delta_2^2 f_1+(\lambda_2+\delta_2)f_2$,

$c=\delta_2 f_2$,

$\lambda_3=\dfrac{-2c}{b\pm\sqrt{b^2-4ac}}$,

上式分母中"±"号与 b 取同号,于是得到 $x_3=x_2+\lambda_3(x_2-\lambda_1)$,再计算 $f_3=f(x_3)$.

3. 控制.如果 x_3 满足 $|\delta|\leqslant\varepsilon_1$ 或 $|f_3|\leqslant\varepsilon_2$($\varepsilon_1,\varepsilon_2$ 为给定精度),则认为迭代收敛,终止迭代,x_3 即为所求,否则执行第 4 步.这里

代法.

对方程 $f(x)=0$,给定迭代函数 $\varphi(x)$,构造多重迭代函数

$$g(x)=\varphi(x)-\frac{f(\varphi(x))}{f'(x)}. \tag{5.6.6}$$

相应迭代法写成

$$\begin{cases} y_k=\varphi(x_k), \\ x_{k+1}=y_k-\dfrac{f(y_k)}{f'(x_k)}, \end{cases} \quad k=0,1,\cdots \tag{5.6.7}$$

定理 5.6.1 设 $f(x)$在根 x^* 附近为二次连续可导,$f'(x^*)\neq 0$,且 $\varphi(x)$为 $p(\geqslant 1)$阶迭代函数,则 $g(x)$为 $p+1$ 阶迭代函数.

根据定理结论可知多重迭代法(5.6.7)为 $p+1$ 阶收敛,如果取

$$\varphi(x)=x-\frac{f(x)}{f'(x)}, \tag{5.6.8}$$

即 $\varphi(x)$ 为牛顿法迭代函数,此时 $p=2$,于是多重迭代公式(5.6.7)可写成

$$\begin{cases} y_k=x_k-\dfrac{f(x_k)}{f'(x_k)}, \\ x_{k+1}=y_k-\dfrac{f(y_k)}{f'(x_k)}, \end{cases} \quad k=0,1,\cdots. \tag{5.6.9}$$

它的收敛阶为 3. 对式(5.6.9)每步迭代只需计算两次 f 值及一次导数值 f'.

利用牛顿迭代法还可构造另一个具有更高阶的多重迭代法

$$\begin{cases} y_k=x_k-\dfrac{f(x_k)}{f'(x_k)}, \\ x_{k+1}=y_k-\dfrac{f(y_k)(y_k-x_k)}{2f(y_k)-f(x_k)}, \end{cases} \quad k=0,1,\cdots \tag{5.6.10}$$

它的迭代函数

$$g(x)=\varphi(x)-\frac{f(\varphi(x))(\varphi(x)-x)}{2f(\varphi(x))-f(x)}, \tag{5.6.11}$$

$$\delta = \begin{cases} |x_3 - x_2|, & \text{当 } |x_3| < \gamma, \\ \dfrac{|x_3 - x_2|}{|x_3|}, & \text{当 } |x_3| \geqslant \gamma, \end{cases} \quad \gamma \text{ 为控制常数.}$$

4. 修改. 如迭代次数达到指定的次数 N,则认为迭代不收敛,否则以 $(x_1, x_2, x_3, f_1, f_2, f_3, \lambda_3)$ 分别代替 $(x_0, x_1, x_2, f_0, f_1, f_2, \lambda_2)$,转 2 继续迭代.

5.6 多重迭代法

从迭代法加速得到的斯蒂芬森法(5.3.3),实际上是一种多重迭代法. 一般情形,若给定两个迭代函数 $\varphi(x)$ 及 $\psi(x)$,可构造**多重迭代法**

$$\begin{cases} y_k = \varphi(x_k), \\ x_{k+1} = \psi(y_k), \end{cases} \tag{5.6.1}$$

$k=0,1,\cdots$. 它等价于不动点迭代法

$$x_{k+1} = g(x_k) \quad (k = 0,1,\cdots), \tag{5.6.2}$$

其中迭代函数

$$g(x) = \psi(\varphi(x)). \tag{5.6.3}$$

如果 $x \to x^*$ 时有

$$\begin{cases} \varphi(x) - x^* = O(|x - x^*|^{p_1}), \\ \psi(x) - x^* = O(|x - x^*|^{p_2}), \end{cases} \tag{5.6.4}$$

则

$$g(x) - x^* = O(|\varphi(x) - x^*|^{p_2}) = O(|x - x^*|^{p_1 p_2}), \tag{5.6.5}$$

即迭代函数 $\varphi(x), \psi(x)$ 的收敛阶分别为 p_1 及 p_2 时,$g(x) = \psi(\varphi(x))$ 的收敛阶不低于 $p_1 p_2$,如果构造的多重迭代收敛阶不高于 $p_1 p_2$ 就没有实用价值,为使构造的多重迭代法具有实用价值,就要使其收敛阶高于 $p_1 p_2$. 下面考虑具有加速效果的多重迭

其中 $\varphi(x)$ 是表达式(5.6.8). 这个迭代公式计算量与式(5.6.9)相当,但收敛阶为4.

定理 5.6.2 设 $f(x)$ 于根 x^* 附近三次连续可导,且 $f'(x^*)\neq 0$,则多重迭代法(5.6.10)是四阶收敛的.

例 5.6.3 用多重迭代法(5.6.9)及(5.6.10)计算 $f(x)=x^3-2x-5=0$ 在区间(2,3)内的实根.

解 取 $x_0=2$,此方程的一个根 $x^*=2.09455148154$,用式(5.6.9)及式(5.6.10)计算,结果见表5.3.

表 5.3

迭代次数	三阶迭代法(5.6.9)		四阶迭代法(5.6.10)	
k	x_k	$\varepsilon_k=x_k-x^*$	x_k	$\varepsilon_k=x_k-x^*$
0	2		2	
1	2.093900000	-0.651×10^{-3}	2.0945632798	0.118×10^{-4}
2	2.0945514813	-0.240×10^{-9}	2.0945514815	-0.400×10^{-10}
3	2.0945514815	-0.400×10^{-10}		

从表中结果看到三阶与四阶迭代法只要迭代2,3步则可得到精确度很高的根,而要达到同样精度的根用弦截法至少要迭代6次. 可见这是两种比较有效的迭代法.

5.7 重根计算

前述牛顿法、弦截法和多重迭代法关于收敛阶的结论都假定了方程为单根,如果 x^* 是 $f(x)=0$ 的 m 重根($m>1$),则结论不成立. 此时,$f(x)=(x-x^*)^m g(x)$,$g(x^*)\neq 0$,$m\geqslant 2$,且

$$f(x^*)=f'(x^*)=\cdots=f^{(m-1)}(x^*)=0,\quad f^{(m)}(x^*)\neq 0. \tag{5.7.1}$$

若用牛顿法(5.4.3)求根,其迭代函数

$$\varphi(x)=x-\frac{f(x)}{f'(x)},\quad \varphi'(x^*)=1-\frac{1}{m}\neq 0,$$

故用牛顿法求重根收敛阶为1,若取

$$\varphi(x)=x-m\frac{f(x)}{f'(x)},$$

则 $\varphi'(x^*)=0$. 由此构造的迭代法

$$x_{k+1}=x_k-m\frac{f(x_k)}{f'(x_k)}\quad (k=0,1,\cdots)\tag{5.7.2}$$

用于计算 m 重根,其收敛阶仍然是2,但必须事先知道根 x^* 的重数 m. 通常 m 并不知道,为了构造不受 m 影响的迭代方法,令 $u(x)=\dfrac{f(x)}{f'(x)}$,若 x^* 是 $f(x)=0$ 的 m 重根,则

$$u(x)=\frac{(x-x^*)g(x)}{mg(x)+(x-x^*)g'(x)},$$

故 x^* 是 $u(x)=0$ 的单根. 对它用牛顿法得

$$x_{k+1}=x_k-\frac{u(x_k)}{u'(x_k)},$$

$$u'(x_k)=1-\frac{f''(x_k)}{f'(x_k)}u(x_k).\tag{5.7.3}$$

其迭代函数

$$\varphi(x)=x-\frac{u(x)}{u'(x)}=x-\frac{f(x)f'(x)}{[f'(x)]^2-f(x)f''(x)}.$$

于是式(5.7.3)可改写成

$$x_{k+1}=x_k-\frac{f(x_k)f'(x_k)}{[f'(x_k)]^2-f(x_k)f''(x_k)},\tag{5.7.4}$$

它的收敛阶为2. 它要计算 $f''(x)$,当 $f''(x)$ 较复杂时计算量较大,可改用弦截法求 $u(x)=0$ 的根,其迭代程序为

$$x_{k+1}=x_k-\frac{x_k-x_{k-1}}{u(x_k)-u(x_{k-1})},\quad k=0,1,\cdots\tag{5.7.5}$$

例 5.7.1 方程 $x^4-4x^2+4=0$ 的根 $x^*=\sqrt{2}$是二重根，用牛顿法及迭代公式(5.7.2)，(5.7.4)求根.

解 因 $f(x)=x^4-4x^2+4, f'(x)=4x^3-8x, f''(x)=12x^2$，分别求出三种方法的迭代公式如下：

(1) 牛顿法 $x_{k+1}=x_k-\dfrac{x_k^2-2}{4x_k}$.

(2) 迭代公式(5.7.2)为 $x_{k+1}=x_k-\dfrac{x_k^2-2}{2x_k}$.

(3) 迭代公式(5.7.4)为 $x_{k+1}=x_k-\dfrac{x_k(x_k^2-2)}{x_k^2+2}$.

取初始近似 $x_0=1.5$，用上述三种方法计算结果见表 5.4.

表 5.4

k	x_k	方法(1)	方法(2)	方法(3)
1	x_1	1.458333333	1.416666667	1.411764706
2	x_2	1.436607143	1.414215686	1.414211438
3	x_3	1.425497619	1.414213562	1.414213562

计算三步，方法(2)，方法(3)均达到 10 位有效数字，而用牛顿法直接计算只有 2 位有效数字，若要达到 10 位有效数字必须计算 30 步.

5.8 代数方程求根与迭代法

5.8.1 引言与多项式求值

当 $f(x)$是由式(5.1.2)给出的代数多项式时，方程 $f(x)=0$ 就称为**代数方程**，按代数学基本定理. 此方程在复数域中有 n 个根 $x_1^*,\cdots,x_n^*$，且 $f(x)=a_0(x-x_1^*)\cdots(x-x_n^*)$. 当系数 $a_0,a_1,\cdots,a_n$ 为实数时，如果方程有复根，则它必共轭地成对出现. 由于多项

式有很多特殊性质,求根要求不同,有时只要知道根的界,有时只要判断右半平面有没有根,有时则要求全部根,因此,代数方程求根除了可用一般函数方程求根方法,还有很多特殊方法.

在各种求根方法中,计算工作量主要指求 $f(x)$ 值的多少.如用牛顿法求根,每步要计算 $f(x_k)$ 及 $f'(x_k)$ 各一次.对多项式方程

$$f(x) = a_0x^n + a_1x^{n-1} + \cdots + a_{n-1}x + a_n = 0, \quad (5.8.1)$$

为了计算 $f(x_k)$的值,可用 $x-x_k$ 除 $f(x)$得

$$f(x) = q(x)(x-x_k) + f(x_k), \quad (5.8.2)$$

其余式 $f(x_k)$就是多项式 $f(x)$在点 x_k 的值.商 $q(x)$可表示为

$$q(x) = b_0x^{n-1} + \cdots + b_{n-2}x + b_{n-1}.$$

比较式(5.8.2)两端 x 同次幂系数,则得

$$\begin{cases} b_0 = a_0 \\ b_i = a_i + x_k b_{i-1}, \quad (i = 1,\cdots,n). \\ f(x_k) = b_n, \end{cases} \quad (5.8.3)$$

用这个公式计算多项式值的方法称为**秦九韶算法**,它只需用 n 次乘法和 n 次加法运算,因此计算量小,结构紧凑,便于编制程序.应用这个方法求 $f'(x_k)$,只要对式(5.8.2)两端求导,则可得 $f'(x_k)=q(x_k)$,也就是对 $q(x)$继续用秦九韶方法.令

$$\begin{cases} q(x) = p(x)(x-x_k) + q(x_k), \\ p(x) = c_0x^{n-2} + \cdots + c_{n-3}x + c_{n-2}, \end{cases} \quad (5.8.4)$$

比较(5.8.4)两端系数则得

$$\begin{cases} c_0 = b_0 \\ c_i = b_i + x_k c_{i-1}, \quad (i = 1,\cdots,n-1). \\ f'(x_k) = q(x_k) = c_{n-1}, \end{cases} \quad (5.8.5)$$

5.8.2 牛顿法与拉盖尔迭代法

根据上述多项式求值方法(5.8.3)和(5.8.5)可知,$f(x_k)=b_n$,$f'(x_k)=c_{n-1}$,于是用牛顿法求方程(5.8.1)根的计算公式为

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = x_k - \frac{b_n}{c_{n-1}} \quad (k = 0,1,\cdots). \tag{5.8.6}$$

由 x_k 算出 x_{k+1} 的步骤是：

1. 令 $b_0=c_0=a_0$；
2. 对 $i=1,\cdots,n-1$ 做
$$b_i = a_i + x_k b_{i-1}, \quad c_i = b_i + x_k c_{i-1};$$
3. 由 $b_n=a_n+x_k b_{n-1}$，得 $x_{k+1}=x_k-\dfrac{b_n}{c_{n-1}}$.

整个求解过程只要对 $k=0,1,\cdots$，算到 x_{k+1} 满足精度要求为止.

除了牛顿法，还可用弦截法及多重迭代法求多项式方程的根，但这些方法与牛顿法都必须给出足够好的初始近似 x_0，因此这些方法都不是特别好用，于是选取某种具有全局收敛性的方法显得更为重要. 具有这种特性的一个较好方法是拉盖尔迭代法，它的计算公式是

$$x_{k+1} = x_k - \frac{nf(x_k)}{f'(x_k) \pm \sqrt{H(x_k)}} \quad (k = 0,1,\cdots), \tag{5.8.7}$$

其中

$$H(x) = (n-1)[(n-1)(f'(x))^2 - nf(x)f''(x)], \tag{5.8.8}$$

这里 n 是多项式 $f(x)$ 的次数. 式(5.8.7)分母中符号应选得与 $f'(x_k)$ 符号相同，使得 $|x_{k+1}-x_k|$ 的值尽量小. 拉盖尔法每步要计算 $f(x_k)$，$f'(x_k)$ 和 $f''(x_k)$，对于 $f''(x_k)$ 的计算，由式(5.8.2)和式(5.8.4)可得 $f''(x_k)=2p(x_k)$，而计算 $p(x_k)$ 可在式(5.8.5)的基础上类似得到

$$d_0 = c_0, \quad d_i = d_{i-1}x_k + c_i \quad (i = 1,2,\cdots,n-2),$$
$$p(x_k) = d_{n-2}.$$

可以证明,拉盖尔迭代法对于单根(实或复)是三阶收敛的,且对于实根,任给初始近似 x_0,迭代序列都是收敛的. 设方程的 n 个实根排列成 $x_1^* \leqslant x_2^* \leqslant \cdots \leqslant x_n^*$,如果初始近似 $x_0 \in (x_{j-1}^*, x_j^*)$,则拉盖尔法收敛于根 x_{j-1}^* 或 x_j^* $(j=2,\cdots,n)$; 若 $x_0 < x_1^*$ 或 $x_0 > x_n^*$,则分别收敛于 x_1^* 及 x_n^*. 这些结论对复根不再成立,但经验表明对复根全局收敛性质还是好的.

注意,对于实系数多项式 $f(x)$,当 x_k 为实数时,$f(x_k)$,$f'(x_k)$和 $f''(x_k)$都是实数,这意味着从实初始近似 x_0 出发,牛顿法、弦截法和多重迭代法都不能收敛于复根,可是拉盖尔法即使 x_k 是实的,x_{k+1}也可以是复的,因为有 $H(x_k)<0$,因此这个方法用实初始近似 x_0 也可收敛到复根.

用迭代法求出 $f(x)=0$ 的一个根 x^*,则可通过因式分解得到

$$f(x) = (x - x^*)g_1(x),$$

其中

$$g_1(x) = b_0 x^{n-1} + \cdots + b_{n-2}x + b_{n-1}.$$

于是方程剩下的根也是 $g_1(x)=0$ 的根,重复此过程即可求得方程的全部根. 应特别指出的是这种算法的舍入误差影响逐次商多项式的零点,使它们越来越偏离 $f(x)$的零点. 为保险起见,可对每一个根用原始多项式 $f(x)$作最后一次迭代.

5.9 根模的界与实根隔离

5.9.1 根模的上下界

有的问题只要求知道根模的界,如特征值估计; 而知道根模的界也就得到有根区域. 为了估计根模的界,可把方程(5.8.1)改写成标准形式

$$f(x) = x^n + a_1 x^{n-1} + \cdots + a_{n-1} x + a_n = 0. \quad (5.9.1)$$

它的 n 个根 $x_i(i=1,\cdots,n)$ 的绝对值 $|x_i|$ 称为根模. 如有常数 $0<A<B$,使

$$A \leqslant |x_i| \leqslant B \quad (i = 1,\cdots,n)$$

成立,则称 A 为根模下界,B 为根模上界. 求根模上界有以下定理.

定理 5.9.1 对方程(5.9.1)的所有根 x_i,有

$$|x_i| \leqslant \max\{|a_1|+1,\cdots,|a_{n-1}|+1,|a_n|+1\},$$

或

$$|x_i| \leqslant \max\left\{1,\sum_{j=1}^{n}|a_j|\right\}.$$

这种求根模的方法很简单,但求出的上界通常太大. 为求得更好的上界,可令

$$u(\rho) = \rho^n - |a_1|\rho^{n-1} - \cdots - |a_{n-1}|\rho - |a_n|.$$

显然 $|f(x)| \geqslant u(|x|)$. 由于 $u(0) = -|a_n|$,$u(+\infty)>0$,方程 $u(\rho)=0$ 在 $(0,+\infty)$ 有惟一正根 ρ_0,所以当 $\rho\in(\rho_0,+\infty)$ 时 $u(\rho)>0$. 若存在 ρ_1,使 $u(\rho_1)>0$,则当 $|x|\geqslant\rho_1$ 时 $|f(x)|\geqslant u(|x|)>0$,这时方程(5.9.1)无根,故 ρ_1 是根模的一个上界.

例 5.9.2 估计方程 $f(x)=x^3-2x-5=0$ 的根模的上界.

解 由定理 5.9.1 有 $|x_i|\leqslant 6$,6 是根模的一个上界. 为了求更好的上界,可取 $\rho_1=3$,则 $u(3)=16>0$,故 $|x_i|\leqslant 3$,3 是一个更好的上界. 若取 $\rho_1=2.1$,因 $u(2.1)=0.061>0$,故 $|x_i|\leqslant 2.1$ 是一个更好的根模上界.

为了求根模下界,可令

$$L(\sigma) = \sigma^n + |a_1|\sigma^{n-1} + \cdots + |a_{n-1}|\sigma - |a_n|.$$

显然有 $|f(x)| \geqslant -L(|x|)$. 当 $\sigma>0$,$L(\sigma)$ 是单调增函数时,$L(0)<0$,$L(+\infty)>0$,故 $L(\sigma)$ 只有一个正根 $\sigma_0>0$. 当 $0\leqslant\sigma<\sigma_0$ 时 $L(\sigma)<0$,若存在 $\sigma_1<\sigma_0$,使 $L(\sigma_1)<0$,则当 $|x|\leqslant\sigma_1$,有 $L(|x|)<0$.

因此$|f(x)|\geqslant -L(|x|)>0$,表明$|x|\leqslant\sigma_1$时方程(5.9.1)没有根,故$|x_i|\geqslant\sigma_1$,$\sigma_1$即是根模的一个下界,而根模$|x_i|$的最大下界为$\sigma_0$.

如果$f(x)\neq 0$,令$z=\dfrac{1}{x}$,则z满足另一个n次代数方程

$$a_n z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + 1 = 0.$$

若$a_n\neq 0$,以a_n除之即得

$$g(z) = z^n + \frac{a_{n-1}}{a_n} z^{n-1} + \cdots + \frac{a_1}{a_n} z + \frac{1}{a_n} = 0.$$

对$g(z)=0$应用定理5.9.1的结论,则得

$$| z_i | = \left|\frac{1}{x_i}\right| \leqslant \max\left\{\left|\frac{a_{n-1}}{a_n}\right| + 1, \cdots, \left|\frac{a_1}{a_n}\right| + 1, \left|\frac{1}{a_n}\right| + 1\right\}$$

$$= b \geqslant 1,$$

$$| z_i | = \left|\frac{1}{x_i}\right| \leqslant \max\left\{1, \sum_{j=1}^{n-1}\left|\frac{a_j}{a_n}\right| + \left|\frac{1}{a_n}\right|\right\} = c \geqslant 1$$

于是有根模下界$|x_i|\geqslant\dfrac{1}{b}$或$|x_i|\geqslant\dfrac{1}{c}$.

例 5.9.3 估计方程$f(x)=x^3-2x-5=0$的根模的下界.

解 由于

$$b = \frac{2}{5} + 1 = \frac{7}{5}, \quad c = 1,$$

故$\dfrac{5}{7}$是根模的一个下界. 1是一个更好的下界,若取$\sigma=1.3$,则$L(1.3)=-0.203<0$,故1.3是一个更好的根模下界. 与例5.9.2的结果合在一起,可知$f(x)=x^3-2x-5=0$的根均在下列圆环内,即$1.3\leqslant|x_i|\leqslant 2.1$.

例 5.9.4 估计方程$f(x)=x^{41}+x^3+1=0$的根模的上、下界.

解 若用定理5.9.1可估得根模上界为2,下界为$\dfrac{1}{2}$,即$\dfrac{1}{2}\leqslant|x_i|\leqslant 2$. 若取$\rho_1=1.018$,计算函数值$u(1.018)=(1.018)^{41}-$

$(1.018)^3-1=0.023>0$,故1.018是一个更好上界.为估计下界,考虑 $L(\sigma)=\sigma^{41}+\sigma^3-1$,取 $\sigma_1=0.9524$,计算 $L(0.9524)=0.9993-1=-0.0007<0$,故 0.9524 是一个更好的下界.于是得到这个方程 41 个根都在以原点为中心的下列圆环内:

$$0.9524 \leqslant |x_i| \leqslant 1.018.$$

这是单位圆邻近很狭窄的一个圆环域.

5.9.2 施图姆序列

为了解决代数方程根的隔离问题,引进以下定义.

定义 5.9.5 实多项式序列

$$f(x)=f_0(x),f_1(x),\cdots,f_m(x) \tag{5.9.2}$$

称为施图姆(Sturm)序列,如果

(1) ξ 是 $f_k(x)$ 的实零点,则对 $k=1,\cdots,m-1$,有 $f_{k-1}(\xi)f_{k+1}(\xi)<0$;

(2) 最后一个多项式 $f_m(x)$ 没有实根.

设 $f_0(x)$ 和 $f_1(x)$ 是两个 x 的多项式,其中 $f_1(x)$ 次数低于 $f_0(x)$.用 $f_1(x)$ 除 $f_0(x)$ 得商 $q_0(x)$ 及余式 $R_2(x)$,$R_2(x)$ 的次数低于 $f_1(x)$.令 $f_2(x)=-R_2(x)$ 为序列的下一个函数,如果 $f_2(x)\not\equiv 0$,用 $f_2(x)$ 除 $f_1(x)$ 得商 $q_1(x)$ 及余式 $R_3(x)$,$R_3(x)$ 次数低于 $f_2(x)$.令 $f_3(x)=-R_3(x)$ 为下一个函数,这样做下去,直到除尽为止.从 f_0,f_1 得出 $m+1$ 个非零多项式 $\{f_0,f_1,\cdots,f_m\}$,次数逐个降低,$f_{m+1}\equiv 0$,这个函数序列中三个相邻函数间的关系是

$$f_k(x)=q_k(x)f_{k+1}(x)-f_{k+2}(x)\quad(k=0,1,\cdots,m-1),$$

$$f_{m+1}(x)\equiv 0.$$

这样得到的序列就是一个施图姆序列,称为**以 f_0 及 f_1 为基的施图姆序列**.通常选 $f_0=f,f_1=f'(x)$ 为基作施图姆序列

$$\{f,f',f_2,\cdots,f_m\}. \tag{5.9.3}$$

定义 5.9.6 实多项式序列(5.9.2),当 $x=a$ 时是一个数列

$\{f_0(a), f_1(a), \cdots, f_m(a)\}$，若两个相邻数符号相反，就说这两个数之间有一次变号，把数列中一切零拿出数列，则数列中各相邻数变号次数之和定义为这个数列的**变号次数**（the number of changes in sign），记作 V_a.

定理 5.9.7（施图姆定理） 设 $f(a) \neq 0, f(b) \neq 0$，以 $f_0 = f$，$f_1 = f'(\cdot)$ 为基的施图姆序列(5.9.3)在点 x 的变号次数为 V_x，则方程 $f(\cdot) = 0$ 在 (a,b) 区间内共有 $V_a - V_b$ 个各不相同的实根，设最后非零函数 $f_m(x)$ 没有实根，则 $f(x)=0$ 的实根都是单根；设 $f_m(x)=0$ 有实根，则这些根都是 $f(x)=0$ 的重根，其重数为 $f_m(x)=0$ 根的重数加 1.

施图姆定理解决了判断给定区间内有没有根，有几个实根的问题. 当 $a=-\infty, b=+\infty$ 时，可解决 $f(x)=0$ 的实根个数；当 (a,b) 内只有一个根，且 $b-a$ 很小时，可解决实根隔离，用二分法容易求出根的足够精确的近似值.

例 5.9.8 $f(x) = 2x^3 - 9x^2 + 11x - 3.5 = 0$.

解 施图姆序列是

$$f_0(x) = f(x) = 2x^3 - 9x^2 + 11x - 3.5,$$

$$f_1(x) = f'(x) = 6x^2 - 18x + 11,$$

$$f_2(x) = \frac{1}{3}(5x - 6),$$

$$f_3(x) = 1.$$

由于求 $f_k(x)$ 时可以相差一个正常数因子，这里 $f_3(x)=1$，故没有重根，相应点变号次数结果如下：

x	$f_0(x)$	$f_1(x)$	$f_2(x)$	$f_3(x)$	V_x
$-\infty$	$-$	$+$	$-$	$+$	3
0	$-$	$+$	$-$	$+$	3
$+\infty$	$+$	$+$	$+$	$+$	0

可见方程只有三个正实根，没有负根. 为进一步将这三个根隔离，可再试算某些点的变号次数，结果如下：

x	$f_0(x)$	$f_1(x)$	$f_2(x)$	$f_3(x)$	V_x
0	−	+	−	+	3
1	+	−	−	+	2
2	−	−	+	+	1
3	+	+	+	+	0

这说明三个根分别在区间(0,1),(1,2),(2,3)中，可用二分法或其他精确化方法把根计算出来.

5.10 伯努利方法

对任何 n 次方程

$$f(x)=x^n+a_1x^{n-1}+\cdots+a_{n-1}x+a_n=0, \qquad (5.10.1)$$

考虑其对应的齐次常系数线性差分方程

$$f(E)u_k=u_{k+n}+a_1u_{k+n-1}+\cdots+a_{n-1}u_{k+1}+a_nu_k=0, \qquad (5.10.2)$$

它的特征方程就是 n 次方程(5.10.1)，方程(5.10.1)模最大的根叫最大根，模最小的根叫最小根. 伯努利方法通常用来求最大根，稍加变动也可用来求方程的最小根.

设方程的最大根为 x_1，其他根 $x_2,\cdots,x_n$ 的模都比 x_1 的模小. 假定给出差分方程的初始条件 $u_0=\cdots=u_{n-2}=0, u_{n-1}=1$，把差分方程(5.10.2)改写成

$$u_{k+n}=-a_1u_{k+n-1}-\cdots-a_{n-1}u_{k+1}-a_nu_k \qquad (5.10.3)$$

由此可逐次算出它的特解 $u_n=-a_1,\cdots$. 这个特解可表示成

$$u_k=c_1x_1^k+c_2x_2^k+\cdots+c_nx_n^k, \quad c_1\neq 0, \qquad (5.10.4)$$

这里 c_i 是由初始条件确定的. 把式(5.10.4)改写成

$$u_k=c_1x_1^k\left[1+\frac{c_2}{c_1}\left(\frac{x_2}{x_1}\right)^k+\cdots+\frac{c_n}{c_1}\left(\frac{x_n}{x_1}\right)^k\right],$$

于是有

$$\frac{u_{k+1}}{u_k}=\frac{c_1x_1^{k+1}\left[1+\sum\limits_{i=2}^{n}\left(\frac{c_i}{c_1}\right)\left(\frac{x_i}{x_1}\right)^{k+1}\right]}{c_1x_1^k\left[1+\sum\limits_{i=2}^{n}\left(\frac{c_i}{c_1}\right)\left(\frac{x_i}{x_1}\right)^{k}\right]}.$$

由于$|x_1|>|x_2|\geqslant\cdots\geqslant|x_n|$,故

$$\lim_{k\to\infty}\frac{u_{k+1}}{u_k}=x_1.$$

这表明:当k充分大时,后面的数u_{k+1}与前面的数u_k的比值是最大根x_1的近似值.以上运算的方法就是求最大实根的伯努利方法.当方程最大根是一对复根(非一对共轭虚根)时,也可用伯努利法(计算公式略).当$|x_2|$(复根为$|x_3|$)很接近$|x_1|$时,其收敛速度很慢,伯努利法就没有多大实用价值.计算方程(5.10.3)特解的方法编成程序后计算很简单,手算时可用计算样板,将方程系数从下到上排列,次序是$-a_1,\cdots,-a_n$,与相应u_k相乘.

例 5.10.1 求 $f(x)=x^3+15x^2+69x+104=0$ 的最大根.

解 计算结果如下表:

	u_k	u_{k+1}/u_k
	0	
−104	0	
−69	1	−15
−15	−15	−10.4
→	156	−9.032
	−1409	−8.4677
	11931	−8.2104
	−97958	−8.0919
	792667	−8.03834
	−6371727	−8.014705
	51067514	−8.005107
	−408800915	−8.001481
	3271012787	

可以看出，当 k 增大时比值 u_{k+1}/u_k 越来越接近最大实根 -8. 由于 $f(x)=x^3+15x^2+69x+104=0$ 可以改写成 $f(x)=(x+8)(x^2+7x+13)=0$，方程最大根是 -8，其他两个根是一对共轭复根 $\frac{-7\pm\sqrt{3}\mathrm{i}}{2}$，它们的模是 $\sqrt{13}\approx 3.606$，$\frac{\sqrt{13}}{8}\approx 0.45$，故收敛速度较慢.

这种方法也可用来求最小根. 设方程(5.10.1)的最小根 $x_n\neq 0$，则 $a_n\neq 0$. 令 $z=\frac{1}{x}$，代入方程(5.10.1)可得 z 满足方程

$$a_n z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + 1 = 0. \qquad (5.10.5)$$

只要对方程(5.10.5)求最大根，则得方程(5.10.1)的最小根. 求最小根的伯努利法也可作为根的精确化的一种有效方法.

5.11 劈因子法

因为二次方程容易求解，故若能找出 n 次多项式 $f(x)$ 的一个二次因式，就等于找到了方程(5.9.1)的一对复根. 劈因子法的基本思想就是从某个近似的二次因子

$$w(x) = x^2 + ux + v$$

出发，用某种迭代过程使之逐步逼近 $f(x)$ 的一个二次因子，从而达到求根目的.

用 $w(x)$ 除 $f(x)$ 得商式 $p(x)$，它是 $n-2$ 次多项式. 一次余式为

$$r(x) = r_1 x + r_2,$$

其中 r_1 和 r_2 两个系数均由 $w(x)$ 的系数 u,v 所确定，因此它们是 u,v 的函数，于是有

$$f(x) = (x^2 + ux + v)p(x) + r_1 x + r_2. \qquad (5.11.1)$$

若 $r_1=r_2=0$，则 $w(x)$ 就是 $f(x)$ 的准确二次因子，一般情形是

$(r_1, r_2) \neq (0,0)$. 这时，可修正 $w(x)$ 的系数 u, v，修正值记作 Δu 与 Δv，修正后的新二次因式为

$$w^*(x) = x^2 + (u + \Delta u)x + (v + \Delta v),$$

它比旧的二次因式 $w(x)$ 更接近 $f(x)$ 对应的准确二次因子. 用 $w^*(x)$ 除 $f(x)$ 得余式

$$r^*(x) = (r_1 + \Delta r_1)x + (r_2 + \Delta r_2),$$

若要求 $r^*(x)=0$，即要求 Δu 与 Δv 引起的 Δr_1 和 Δr_2 满足下列方程组：

$$r_1 + \Delta r_1 = 0, \quad r_2 + \Delta r_2 = 0. \tag{5.11.2}$$

若用微分近似增量，则得

$$\begin{cases} \dfrac{\partial r_1}{\partial u}\Delta u + \dfrac{\partial r_1}{\partial v}\Delta v + r_1 = 0, \\ \dfrac{\partial r_2}{\partial u}\Delta u + \dfrac{\partial r_2}{\partial v}\Delta v + r_2 = 0. \end{cases} \tag{5.11.3}$$

只要求出方程系数 $\dfrac{\partial r_1}{\partial u}, \dfrac{\partial r_2}{\partial u}, \dfrac{\partial r_1}{\partial v}, \dfrac{\partial r_2}{\partial v}$ 及 r_1, r_2，就可从方程组(5.11.3)解出 Δu 及 Δv，从而求得新的二次因子 $w^*(x)$. 计算步骤如下：

1. 计算 r_1 及 r_2. 令

$$p(x) = b_0 x^{n-2} + b_1 x^{n-3} + \cdots + b_{n-2},$$

代入式(5.11.1)，比较同次幂系数，得

$$\begin{cases} b_0 = a_0, \quad b_1 = a_1 - ub_0, \\ b_i = a_i - ub_{i-1} - vb_{i-2}, \\ r_1 = b_{n-1}, \\ r_2 = b_n + ub_{n-1}. \end{cases} \quad i = 2, \cdots, n \tag{5.11.4}$$

2. 计算 $\dfrac{\partial r_1}{\partial u}, \dfrac{\partial r_1}{\partial v}, \dfrac{\partial r_2}{\partial u}, \dfrac{\partial r_2}{\partial v}$. 将式(5.11.1)两端分别对 u, v 微分，可得

$$\begin{cases} xp(x) = -(x^2 + ux + v)\dfrac{\partial p}{\partial u} - \dfrac{\partial r_1}{\partial u}x - \dfrac{\partial r_2}{\partial u}, \\ p(x) = -(x^2 + ux + v)\dfrac{\partial p}{\partial v} - \dfrac{\partial r_1}{\partial v}x - \dfrac{\partial r_2}{\partial v}, \end{cases} \tag{5.11.5}$$

由此式可知，$\dfrac{\partial r_1}{\partial u}$和$\dfrac{\partial r_2}{\partial u}$是以 $w(x)$除 $xp(x)$所得余式的系数，而$\dfrac{\partial r_1}{\partial v}$和$\dfrac{\partial r_2}{\partial v}$是以 $w(x)$除 $p(x)$所得余式的系数. 若令

$$p(x) = (x^2 + ux + v)H(x) + s_1 x + s_2, \tag{5.11.6}$$

其中

$$H(x) = c_0 x^{n-4} + c_1 x^{n-5} + \cdots + c_{n-4},$$

于是

$$\begin{aligned} xp(x) &= (x^2 + ux + v)xH(x) + s_1 x^2 + s_2 x \\ &= (x^2 + ux + v)(xH(x) + s_1) - \\ &\quad (us_1 - s_2)x - vs_1. \end{aligned} \tag{5.11.7}$$

将式(5.11.6)，(5.11.7)与式(5.11.5)比较，可得

$$\begin{cases} -\dfrac{\partial r_1}{\partial v} = s_1, \qquad -\dfrac{\partial r_2}{\partial v} = s_2, \\ \dfrac{\partial r_1}{\partial u} = us_1 - s_2, \quad \dfrac{\partial r_2}{\partial u} = vs_1. \end{cases} \tag{5.11.8}$$

为求得 s_1 及 s_2，可通过比较式(5.11.6)两端系数，得到

$$\begin{cases} c_0 = b_0, \quad c_1 = b_1 - uc_0, \\ c_i = b_i - uc_{i-1} - vc_{i-2}, \quad (i = 2, \cdots, n-2) \\ s_1 = c_{n-3}, \\ s_2 = c_{n-2} + uc_{n-3}, \end{cases} \tag{5.11.9}$$

将求出的 s_1 及 s_2 代入式(5.11.8)，则得方程组(5.11.3)的系数. 由于计算公式(5.11.9)与式(5.11.4)相似，故编制的程序简单.

3. 由式(5.11.3)解出 Δu 及 Δv，若 $|\Delta u| \leqslant \varepsilon$，$|\Delta v| \leqslant \varepsilon$ 成立，则迭代停止，转第 4 步，否则令 $u := u + \Delta u$，$v := v + \Delta v$，再转第 1

步重新计算.

4. 求 $w^*(x)=x^2+ux+v$ 的解

$$x=\frac{-u\pm\sqrt{u^2-4v}}{2},\tag{5.11.10}$$

上述步骤描述的算法叫做**劈因子法**，也称**贝尔斯托(Bairstow)方法**. 对有一对共轭虚根的二次因式，它的收敛性、收敛速度与求复根的牛顿法等价，但它不用进行复数运算.

例 5.11.1 用劈因子法求 $f(x)=x^4+x^3+5x^2+4x+4=0$ 的近似根.

解 取尾部作为近似二次因式，即

$$w_0(x)=x^2+0.8x+0.8,$$

按第 1,2 步的式(5.11.4),(5.11.9)和式(5.11.8)求出方程组系数及自由项，得到方程组

$$\begin{cases}-0.608+3.72\Delta u-0.6\Delta v=0,\\-0.768+0.48\Delta u+3.24\Delta v=0.\end{cases}$$

由此解出

$$\Delta u=0.19697,\quad \Delta v=0.20786.$$

迭代一次得到

$$w_1(x)=x^2+0.99697x+1.00786,$$

由式(5.11.10)求出近似解

$$x=-0.49849\pm0.87142\mathrm{i},$$

准确的二次因子是

$$w^*(x)=x^2+x+1,$$

解是 $x^*=-0.5\pm0.866\mathrm{i}$.

若要得到更精确的二次因子，可从 $w_1(x)$ 出发按同样方法再迭代下去，直到满足精度要求为止.

另一种简单的劈因子法是用 $xw(x)$ 除 $f(x)$ 得二次余式 $R(x)$，于是有

$$f(x) = xw(x)S(x) + R(x), \qquad (5.11.11)$$

其中 $S(x)$是商式,余式

$$R(x) = r_0x^2 + r_1x + r_2.$$

设 $r_0 \neq 0$,则 $R(x)$可改写成

$$R(x) = r_0\left(x^2 + \frac{r_1}{r_0}x + \frac{r_2}{r_0}\right) = r_0\tilde{w}(x),$$

其中

$$\tilde{w}(x) = x^2 + \frac{r_1}{r_0}x + \frac{r_2}{r_0} = x^2 + \tilde{u}x + \tilde{v},$$

$$\tilde{u} = \frac{r_1}{r_0}, \quad \tilde{v} = \frac{r_2}{r_0}.$$

于是由式(5.11.11)有

$$f(x) = xw(x)S(x) + r_0\tilde{w}(x).$$

设 $f(x)=0$ 没有零根,那么 $f(x)$有二次因式的必要充分条件是 $w(x)=\tilde{w}(x)$. 把从 $w(x)$求出 $\tilde{w}(x)$的过程当作一个迭代过程,如果这个迭代过程收敛,则极限二次式就是 $f(x)$的二次因式,这种迭代法叫林士谔法. 使用这种方法,两次迭代比一次劈因子法的工作量小,而效果大致与一次劈因子法相当,因此计算程序比劈因子法简单得多. 缺点是这个方法并不是都收敛的.

5.12 复根的隔离

使用 Ruth 定理,可以解决判断代数方程在右半平面有没有根,有几个根的问题. 设给定 n 次实系数方程为

$$f(z) = z^n + a_1z^{n-1} + \cdots + a_{n-1}z + a_n = 0, \qquad (5.12.1)$$

它有 n 个根,可能是实根,也可能是复根,表示为 $z_1, z_2, \cdots, z_n$. 如果有两个根相同就称为二重根. 利用这些根,可把方程(5.12.1)改写成

$$f(z) = (z - z_1)\cdots(z - z_n). \qquad (5.12.2)$$

设虚轴上没有方程(5.12.1)的根，则

$$f(\mathrm{i}t) \neq 0, \quad \mathrm{i}t - z_k \neq 0 \quad (-\infty < t < +\infty).$$

由于复数乘积的辐角等于各因子辐角的和，所以有

$$\arg f(\mathrm{i}t) = \sum_{k=1}^{n} \arg(\mathrm{i}t - z_k).$$

当 t 从 $+\infty$ 变到 $-\infty$ 时，把 t 的角度函数 $\phi(t)$ 的增量记作 $\Delta\phi$，则

$$\Delta\phi = \phi(-\infty) - \phi(+\infty),$$

$$\Delta\arg f(\mathrm{i}t) = \sum_{k=1}^{n} \Delta\arg(\mathrm{i}t - z_k).$$

设根在右半平面上，则矢量 $\mathrm{i}t - z_k$ 是从 z_k 到虚轴上动点 $\mathrm{i}t$ 的矢量，当 t 从 $+\infty$ 变到 $-\infty$，它的辐角从 $\frac{\pi}{2}$ 增加到 $\frac{3}{2}\pi$：

$$\Delta\arg(\mathrm{i}t - z_k) = \pi.$$

设 z_k 在左半平面，则

$$\Delta\arg(\mathrm{i}t - z_k) = -\pi.$$

设右半平面有 r 个根，左半平面有 l 个根，$l + r = n$，于是有

$$\begin{aligned}\frac{1}{\pi}\Delta\arg f(\mathrm{i}t) &= \sum_{k=1}^{n} \frac{1}{\pi}\Delta\arg(\mathrm{i}t - z_k) \\ &= r - l = r - (n - r) = 2r - n.\end{aligned}$$

在右半平面根的个数为

$$r = \frac{1}{2}\left[n + \frac{1}{\pi}\Delta\arg f(\mathrm{i}t)\right]. \qquad (5.12.3)$$

为求 r，记

$$\delta f(z) = \frac{1}{\pi}\Delta\arg f(\mathrm{i}t),$$

$$f(\mathrm{i}t) = \mathrm{i}^n[f_0(t) - \mathrm{i}f_1(t)] = \mathrm{i}^n w(t),$$

其中

$$\begin{cases} f_0(t) = t^n - a_2 t^{n-2} + a_4 t^{n-4} - \cdots, \\ f_1(t) = a_1 t^{n-1} - a_3 t^{n-3} + a_5 t^{n-5} - \cdots, \end{cases} \tag{5.12.4}$$

$$w(t) = f_0(t) - \mathrm{i} f_1(t).$$

由于

$$\arg f(\mathrm{i}t) = \arg(\mathrm{i}^n) + \arg w(t),$$

i^n 为常数，$\Delta \mathrm{i}^n = 0$，故

$$\delta f(z) = \frac{1}{\pi} \Delta \arg f(\mathrm{i}t) = \frac{1}{\pi} \Delta \arg w(t).$$

当 t 从 $+\infty$ 变到 $-\infty$ 时，研究 $\phi(t) = \arg w(t)$ 的增量可改为研究 $\cot\phi(t) = -\dfrac{f_0(t)}{f_1(t)}$ 的变号情况. 设 $\cot\phi$ 在虚轴上从正变到负共有 α 次，从负变到正共有 β 次，则

$$\delta f(z) = \alpha - \beta.$$

当 $\cot\phi$ 从正变到负时，$\{f_0(t), f_1(t)\}$ 从异号变为同号，损失一次变号；当 $\cot\phi$ 从负变到正时，$\{f_0(t), f_1(t)\}$ 增加一次变号. 以 $f_0(t), f_1(t)$ 为基作施图姆序列

$$\{f_0(t), f_1(t), \cdots, f_m(t)\},$$

令

$$V_t = V\{f_0(t), f_1(t), \cdots, f_m(t)\},$$

则当 t 从 $+\infty$ 变到 $-\infty$ 时，施图姆序列损失的变号次数是由 $f_0(t)$ 的实零点引起的，中间函数的实零点对变号次数的改变无影响. 由于已设 $f(\mathrm{i}t) \neq 0$，所以 $f_m(t)$ 没有实零点，它是

$$V_\infty - V_{-\infty} = \alpha - \beta = \delta f(z),$$

将其代入式(5.12.3)，则得

$$r = \frac{1}{2}(n + V_\infty - V_{-\infty}). \tag{5.12.5}$$

定理 5.12.1(鲁思(Ruth)定理)　方程(5.12.1)在虚轴及右半平面没有根的充分必要条件是施图姆序列

$$\{f_0(t), f_1(t), \cdots, f_n(t)\}$$

内每个多项式比它前一个多项式低一次，且首项系数都是正数.

定理 5.12.2 设施图姆序列内有 $n+1$ 个非零函数，则每个多项式比它的前一个多项式低一次，方程(5.12.1)在虚轴上没有根，在右半平面根的个数等于首项系数组成的数列的变号次数.

定理 5.12.3 方程(5.12.1)在右半平面没有根，而在虚轴上有 p 个根的充分必要条件是施图姆序列内有 $n-p+1$ 个非零函数

$$\{f_0(t), f_1(t), \cdots, f_{n-p}(t)\},$$

其中每个多项式比它的前一个多项式低一次，首项系数都是正数，而且最后的 p 次多项式有 p 个实零点，这些实根就是方程(5.12.1)在虚轴上的 p 个根的虚部.

定理 5.12.1，定理 5.12.2 和定理 5.12.3 给出了判断虚轴上及右半平面根的个数的方法，为了得到这些结论，要计算以式(5.12.4)的 f_0 及 f_1 为基的施图姆序列. 由于 f_0 与 f_1 一个为偶函数，另一个是奇函数，故施图姆序列中函数交替为奇偶函数，根据除法规则，除去零及交替正负号，可得到计算施图姆序列系数的紧凑表格，叫做 Ruth 表格. 表 5.5 给出 $n=6$ 的 Ruth 表格.

表 5.5

f_0	1	a_2	a_4	a_6
f_1	a_1	a_3	a_5	
f_2	D_0	D_1	D_2	
f_3	E_0	E_1		
f_4	F_0	F_1		
f_5	G_0			
f_6	$H_0=F_1$			

其中

$$D_0 = a_2 - \frac{a_3}{a_1}, \quad D_1 = a_4 - \frac{a_5}{a_1}, \quad D_2 = a_6, \cdots$$

计算系数的法则是：

$$本数=上二行右数-\frac{(上二行最左数)\times(上行右数)}{上行最左数}. \tag{5.12.6}$$

$f_k(t)$的实际系数是正负交替的，例如 $f_2(t)=D_0t^4-D_1t^2+D_2$. 如果首项系数都不等于零，根据定理 5.12.2，则此六次方程在右半平面的复根个数是

$$V\{1,a_1,D_0,E_0,F_0,G_0,F_1\}.$$

另外，方程(5.12.1)在虚轴和右半平面没有根的必要条件是各系数都是正数，因此，如果方程有负系数或零系数，则此方程在虚轴或右半平面一定有根.

例 5.12.4 求 $f(z)=z^4+z^2+z+1=0$ 的最右根，即实部最大的根.

解 由于此方程 z^3 项系数为 0，所以方程在虚轴或右半平面有根. 容易验证此方程在虚轴上无根，故在右半平面有根.

把复平面原点移到 $z_0=1$ 处，即做变换 $u=z-1$，方程变成

$$f(z)=u^4+4u^3+7u^2+7u+4=0.$$

为判断新的复平面右半平面有无根，即 $\mathrm{Re}z>1$ 有无根，作 Ruth 表格：

1	7	4	$f_0(t)=t^4-7t^2+4$
4	7		$f_1(t)=4t^3-7t$
$\frac{21}{4}$	4		$f_2(t)=\frac{21}{4}t^2-4$
$\frac{83}{21}$			$f_3(t)=\frac{83}{21}t$
4			$f_4(t)=4$

因最左列系数均为正，根据定理 5.12.1，方程在 $\mathrm{Re}z\geqslant 1$ 没有根，

故最右根范围在 $0<\mathrm{Re}z<1$ 之间. 选区间 $(0,1)$ 中点 $\frac{1}{2}$ 为复平面新原点，做 $u=z-\frac{1}{2}$ 变换，方程变成

$$f(z)=u^4+2u^3+\frac{5}{2}u^2+\frac{5}{2}u+\frac{29}{16}=0.$$

作 Ruth 表格：

$$\begin{array}{ccc}
1 & \frac{5}{2} & \frac{29}{16} \\
2 & \frac{5}{2} & \\
\hline
\frac{5}{4} & \frac{29}{16} & \\
-\frac{4}{10} & & \\
\frac{29}{16} & &
\end{array}$$

最左列系数有两次变号，故在 $\mathrm{Re}z>\frac{1}{2}$ 时有一对共轭复根，最右根范围缩小到 $\frac{1}{2}<\mathrm{Re}z<1$，这样用对分法做下去，可求出最右根的实部近似值 $x_0=0.5474$. 求 $f(z)$ 在 x_0 的展开式 $f(z)=u^4+2.1896u^3+2.79788u^2+2.75091u+1.93684$. 作 Ruth 表格：

$$\begin{array}{ccll}
1 & 2.79788 & 1.93684 & \\
2.1896 & 2.75091 & & \\
\hline
1.54153 & 1.93684 & & f_2(t)=1.54153t^2-1.93684 \\
-0.00019 & & & f_3(t)=-0.00019t\approx 0 \\
1.93684 & & &
\end{array}$$

最左列系数仍有两次变号，但 $f_3(t)\approx 0$，说明

$$f_2(t) = 1.54153t^2 - 1.93684 = 0$$

的根是虚部近似根，而 $x_0=0.5474$ 已是实部近似值，由 $f_2(t)=0$ 可得

$$t = \pm\sqrt{\frac{1.93684}{1.54153}} \approx \pm 1.12087.$$

于是所求最右根的近似值是 $0.5474\pm 1.12087i$.

运用 Ruth 表格和 $f(z)$ 在点 x_k 的展开式，可隔离复根达到相当的精度，将上述方法编成程序可得到复根相当精确的初始近似，再用收敛快的精确化方法就可求得方程(5.12.1)的复根.

5.13 复多项式的圆盘迭代法

对复多项式

$$p(z) = \prod_{j=1}^{n}(z-\xi_j) = 0 \tag{5.13.1}$$

求根的圆盘迭代法，最早是由 Gargantini 和 Henrici 等人于 1969 年提出的，近十多年有不少发展，这类方法具有收敛速度快，能求出全部根，并且能并行计算等优点，因此，求多项式零点的圆盘算法是一类重要方法.

设 $z\in\mathbb{C}$，$r\in\mathbb{R}$，$r\geqslant 0$，则有界闭集

$$Z = [z;r] = \{z' \in \mathbb{C} \mid |z'-z| \leqslant r\} \tag{5.13.2}$$

称为$\mathbb{C}$中的一个圆盘. z 为圆盘 Z 的中心，记作 $\mathrm{mid}Z=z$，r 为圆盘 Z 的半径，记为 $\mathrm{rad}Z=r$. 当 $r=0$，定义$[z;0]=z$ 为点圆. 设 $Z_i=[z_i;r_i](i=1,2)$，定义圆盘四则运算如下：

$$Z_1 \pm Z_2 = [z_1 \pm z_2; r_1 + r_2],$$

$$Z_1 \cdot Z_2 = [z_1 \cdot z_2; |z_1| r_2 + |z_2| r_1 + r_1 r_2],$$

$$1/Z_2 = \frac{1}{|z_2|^2 - r_2^2}[\bar{z}_2; r_2], \quad 0 \notin Z_2,$$

$$Z_1/Z_2 = Z_1 \cdot 1/Z_2, \quad 0 \notin Z_2,$$

其中 $\bar{z}_2$ 表示 z_2 的共轭复数,$|\cdot|$表示复数模.

作为一种特殊情形,假定 $p(z)$的 $n-1$ 个零点位置已知,在一定条件下可确定另一个零点的位置.

定理 5.13.1 设 $W_2,\cdots,W_n$ 为 $n-1$ 个已知圆盘,若方程(5.13.1)的根 $\xi_i \in W_i(i=2,\cdots,n)$,$z_0 \notin W_i(i=2,\cdots,n)$,且满足条件

$$\frac{p'(z_0)}{p(z_0)} = c_1, \tag{5.13.3}$$

则 $p(z)$的另一零点

$$\xi_1 \in W_1 = z_0 - \frac{1}{c_1 - Z_1},$$

其中

$$Z_1 = \sum_{j=2}^{n} \frac{1}{z_0 - W_j}.$$

实际上,由方程(5.13.1)可知

$$c_1 = \frac{p'(z_0)}{p(z_0)} = \sum_{j=1}^{n} \frac{1}{z_0 - \xi_j},$$

因此有

$$\frac{1}{z_0 - \xi_1} = c_1 - \sum_{j=2}^{n} \frac{1}{z_0 - \xi_j}.$$

于是,由 $\xi_i \in W_i(i=2,\cdots,n)$有

$$\xi_1 = z_0 - \frac{1}{c_1 - \sum_{j=2}^{n} \frac{1}{z_0 - \xi_j}} \in z_0 - \frac{1}{c_1 - \sum_{j=2}^{n} \frac{1}{z_0 - W_j}} = W_1.$$

因为 $z_0 \notin W_j(j=2,3,\cdots)$,故 Z_1 有意义,它仍然是一个圆盘.如果 z_0 接近 ξ_1,则 $p(z_0)$将很小,此时 $p(z_0)Z_1$ 是中心接近原点的半径很小的圆盘,在这种情况下

$$W_1 = z_0 - \frac{p(z_0)}{p'(z_0) - p(z_0)Z_1} \tag{5.13.4}$$

与点牛顿修正非常接近，用这种思想去建立计算方法，可看成古典牛顿法的圆盘变形. 如果假定 $W_1^{(0)}=[z^{(0)};r^{(0)}]$已给定，且 $W_1^{(0)}$ 只包含 $p(z)$的一个零点 ξ_1，而开圆盘 $|z-z_0|<\rho_0,\rho_0>r^{(0)}$ 不包含 $p(z)$其他零点，定义集合 $U=\{z\,|\,|z-z^{(0)}|\geqslant\rho_0\}$，则 U 包含了 $p(z)$的所有其他零点 $\xi_2,\cdots,\xi_n$，且有

$$\xi_j\in W_j\subseteq U\quad(j=2,\cdots,n).$$

于是由圆盘运算性质知

$$Z_1=\sum_{j=2}^{n}\frac{1}{z^{(0)}-W_j}\subseteq\frac{n-1}{z^{(0)}-U}.\tag{5.13.5}$$

由此，应用定理 5.13.1 就可构造确定 $p(z)$的零点 ξ_1 的圆盘迭代法. 取 $W_1^{(0)}$ 为初始圆盘，则

$$z^{(0)}=\mathrm{mid}W_1^{(0)},\quad Z_1^{(0)}=\frac{n-1}{z^{(0)}-U},$$

由此可构造圆盘迭代算法：

$$\begin{cases}z^{(k)}=\mathrm{mid}W_1^{(k)}, & Z_1^{(k)}=\dfrac{n-1}{z^{(k)}-U}\\ W_1^{(k+1)}=z^{(k)}-\dfrac{1}{q(z^k)-Z_1^{(k)}}, & q(z^k)=\dfrac{p'(z^{(k)})}{p(z^{(k)})}\end{cases}\quad k=0,1,\cdots.\tag{5.13.6}$$

此算法也称带误差界限的修正牛顿算法. 若 $\xi_1\in W_1^{(0)}$，则对一切 k 均有 $\xi_1\in W_1^{(k)}$. 使用时要求选择 $W_1^{(0)}$ 包含 ξ_1，但又不包含 $p(z)$的其他零点，因此，这种算法仍是局部收敛的.

定理 5.13.2 对给定的 $W_1^{(0)}=[z^{(0)};r^{(0)}]$，$\xi_1\in W_1^{(0)}$，$W_1^{(0)}$ 不包含 $p(z)$其他零点，若 $r^{(0)}$ 满足

$$6r^{(0)}\leqslant\eta^{(0)},\quad\rho_0=(n-1)\eta^{(0)}\quad(n\geqslant 2),\tag{5.13.7}$$

则由算法(5.13.6)确定的圆盘序列$\{W_1^{(k)}\}$点收敛于 ξ_1，而 $r^{(k)}=\mathrm{rad}W_1^{(k)}$ 满足

$$r^{(k+1)}\leqslant\frac{1}{\eta^{(0)}-2r^{(0)}}(r^{(k)})^2\quad(k=0,1,\cdots),$$

序列$\{r^{(k)}\}$收敛于0,具有平方敛速.

因此,只要包含ξ_1的圆盘半径满足条件(5.13.7),则圆盘序列$\{W_1^{(k)}\}$平方收敛于ξ_1.如果给出n个互不相交的圆盘$W_j^{(0)}(j=1,2,\cdots,n)$,且$\xi_j\in W_j^{(0)}$,则由定理5.13.1及算法(5.10.8)可构造求$p(z)$全部零点的圆盘迭代法.

设$W_j^{(0)}=[z_j^{(0)};\ r_j^{(0)}]$,$\xi_j\in W_j^{(0)}(j=1,\cdots,n)$给定,假定已经算出$W_j^{(k)}=[z_j^{(k)};r_j^{(k)}]$,$\xi_j\in W_j^{(k)}(j=1,\cdots,n)$,则第$k+1$次迭代可构造成

$$\begin{cases} z_j^{(k)} = \mathrm{mid}W_j^{(k)}, \quad Z_j^{(k)} = \displaystyle\sum_{\substack{i=1 \\ i\neq j}}^{n} \frac{1}{z_j^{(k)} - W_i^{(k)}} \\ W_j^{(k+1)} = z_j^{(k)} - \dfrac{1}{q(z_j^{(k)}) - Z_j^{(k)}}, \quad q(z) = \dfrac{p'(z)}{p(z)} \end{cases}$$

$$j = 1,\cdots,n; \quad k = 0,1,\cdots. \tag{5.13.8}$$

算法(5.13.8)是算法(5.13.6)的推广.如果已知$\xi_j\in W_j^{(0)}(j=1,\cdots,n)$,则由式(5.13.8)可同时迭代出$W_j^{(k)}(j=1,\cdots,n)$,且$\xi_j\in W_j^{(k)}(j=1,\cdots,n)$,它具有明显的并行性质.

定理5.13.3 对给定的互不相交圆盘$W_i^{(0)}$,$\xi_i\in W_i^{(0)}(i=1,\cdots,n)$.定义

$$r^{(k)} = \max_{1\leqslant i\leqslant n} \mathrm{rad}W_i^{(k)} \quad (k = 0,1,\cdots).$$

如果$r^{(0)}$满足

$$6r^{(0)} \leqslant \eta^{(0)},$$

其中

$$\eta^{(0)} = \frac{\rho_0}{n-1}, \quad n\geqslant 2,$$

$$\rho_0 = \max_{i\neq j}\{|z|\mid z\in z_j^{(0)} - W_j^{(0)}\}, \quad z_i^{(0)} = \mathrm{mid}W_i^{(0)},$$

则由算法(5.13.8)确定的圆盘序列$\{W_i^{(k)}\}(i=1,\cdots,n)$点收敛于$\xi_i(i=1,\cdots,n)$,且序列$\{r^{(k)}\}$满足

$$r^{(k+1)} \leqslant \frac{1}{\rho_0(\eta^{(0)} - 4r^{(0)})}(r^{(k)})^3 \quad (k = 0,1,\cdots), \qquad (5.13.9)$$

表明序列$\{r^{(k)}\}$收敛于 0 具有立方敛速.

算法(5.13.8)虽然具有收敛快、能同时求解所有零点等优点，但定理条件要求苛刻，具有很强的局部性，故实际使用较困难.

利用复拉格朗日插值公式还可建立不用计算导数的求 $p(z)$ 全部零点的圆盘迭代法. 仍然假定已知 n 个圆盘 $W_i^{(0)} = [z_i^{(0)}; r_i^{(0)}]$，$\xi_i \in W_i^{(0)}\,(i=1,\cdots,n)$，则圆盘迭代程序定义为：

$$W_i^{(k+1)} = z_i^{(k)} - \frac{h_i^{(k)}}{1 - Z_i^{(k)}}, \quad 0 \notin 1 - Z_i^{(k)}, \qquad (5.13.10)$$

$$i = 1,\cdots,n; \quad k = 0,1,\cdots,$$

其中

$$Z_i^{(k)} = \sum_{\substack{j=1 \\ j\neq i}}^{n} \frac{h_j^{(k)}}{z_j^{(k)} - W_i^{(k)}},$$

$$h_i^{(k)} = \frac{p(z_i^{(k)})}{Q'(z_i^{(k)})},$$

$$Q'(z_i) = \prod_{\substack{j=1 \\ j\neq i}}^{n} (z_i - z_j),$$

可以证明算法(5.13.10)在条件

$$\rho_0 > 3(n-1)r^{(0)}$$

下具有立方敛速，它比算法(5.13.8)的条件

$$\rho_0 \geqslant 6(n-1)r^{(0)}$$

减弱，其计算公式也较简单. 若在算法(5.13.10)中将圆盘迭代改为中点序列

$$z_i^{(k+1)} = z_i^{(k)} - \frac{h_i^{(k)}}{1 - \sum\limits_{\substack{j=1 \\ j\neq i}}^{n} \frac{h_j^{(k)}}{z_j^{(k)} - z_i^{(k)}}} \qquad (5.13.11)$$

$$i = 1,\cdots,n; \quad k = 0,1,\cdots,$$

当初值 $z_i^{(0)}(i=1,\cdots,n)$ 充分靠近零点 $\xi_i(i=1,\cdots,n)$ 时，具有立方敛速.

例 5.13.4 求多项式

$$p(z)=z^7+z^5-10z^4-z^3-z+10$$

的零点. 已知零点 $\xi_i\in W_i^{(0)}=[z_i^{(0)};0.3](i=1,2,\cdots,7)$，其中

$$z_1^{(0)}=2.2,$$
$$z_2^{(0)}=1.2+0.1\mathrm{i},$$
$$z_3^{(0)}=-0.8-0.1\mathrm{i},$$
$$z_4^{(0)}=0.1+1.2\mathrm{i},$$
$$z_5^{(0)}=-0.1-0.8\mathrm{i},$$
$$z_6^{(0)}=-1.1+2.2\mathrm{i},$$
$$z_7^{(0)}=-1.1-1.8\mathrm{i},$$

解 用圆盘迭代程序(5.13.10)，可得第一次与第二次迭代的最大圆盘的半径为 $r^{(1)}\approx5.03\times10^{-2}$，$r^{(2)}\approx2.77\times10^{-5}$. 第三次迭代得到下列圆盘：

$$W_1^{(3)}=[2.0000000000000000016-1.11\times10^{-17}\mathrm{i};\ 5.09\times10^{-17}],$$
$$W_2^{(3)}=[1.0000000000000000053+1.29\times10^{-17}\mathrm{i};\ 7.15\times10^{-16}],$$
$$W_3^{(3)}=[-1.0000000000000000003-2.06\times10^{-18}\mathrm{i};\ 3.12\times10^{-17}],$$
$$W_4^{(3)}=[3.06\times10^{-18}+0.99999999999999999\mathrm{i};\ 2.12\times10^{-17}],$$
$$W_5^{(3)}=[1.63\times10^{-18}-0.999999999999999981\mathrm{i};\ 1.09\times10^{-16}],$$
$$W_6^{(3)}=[-1.0000000000000000+2.0000000000000000\mathrm{i};\ 3.61\times10^{-18}],$$
$$W_7^{(3)}=[-1.0000000000000000-2.00000000000000000\mathrm{i};\ 7.92\times10^{-18}].$$

这里最大圆盘半径 $r^{(3)}=r_2^{(3)}\approx7.15\times10^{-16}$. 迭代三次已得到很高精度，这个问题精确的零点是 $\xi_1=2$，$\xi_{2,3}=\pm1$，$\xi_{4,5}=\pm\mathrm{i}$，$\xi_{6,7}=-1\pm2\mathrm{i}$.

5.14 病态代数方程

在代数方程中，有的多项式系数有微小扰动时其根变化很大，这种根对系数变化的敏感性称为**不稳定性**(instability)，这种方程就是病态多项式方程. 通常重根的方程是病态的，有几个根彼此很靠近，则这些根对系数的扰动也是敏感的，有时根看起来分隔得很好，但同样可能是病态的，如例 5.14.1. 若多项式 $p(x)$ 的系数有微小变化，可表示为

$$p_\varepsilon(x) = p(x) + \varepsilon q(x) = 0, \qquad (5.14.1)$$

其中 $q(x) \neq 0$ 是一个多项式. $p_\varepsilon(x)$ 的零点表为 $x_1(\varepsilon), \cdots, x_n(\varepsilon)$，令 $x_1(0), \cdots, x_n(0)$ 为 $p(x)$ 的零点，即 $x_i = x_i(0)(i=1,\cdots,n)$. 将式(5.14.1)对 ε 求导，可得

$$p'(x)\frac{\mathrm{d}x}{\mathrm{d}\varepsilon} + q(x) + \varepsilon q'(x)\frac{\mathrm{d}x}{\mathrm{d}\varepsilon} = 0,$$

$$\frac{\mathrm{d}x}{\mathrm{d}\varepsilon} = \frac{-q(x)}{p'(x) + \varepsilon q'(x)},$$

于是，当 $\varepsilon = 0$ 时有

$$\frac{\mathrm{d}x(0)}{\mathrm{d}\varepsilon} = \frac{-q(x(0))}{p'(x(0))}. \qquad (5.14.2)$$

应用 $x(\varepsilon)$ 的泰勒展开，当 $|\varepsilon|$ 充分小时，得

$$x_k(\varepsilon) \approx x_k - \frac{q(x_k)}{p'(x_k)}\varepsilon \quad (k = 1,\cdots,n). \qquad (5.14.3)$$

这表明了系数有微小变化 ε 时引起根变化的情况，当 $x_i(\varepsilon) - x_i$ 很大时，就称方程是病态的或不稳定的.

例 5.14.1 多项式

$$\begin{aligned} p(x) &= (x-1)(x-2)\cdots(x-7) \\ &= x^7 - 28x^6 + 322x^5 - 1960x^4 + \\ &\quad 6769x^3 - 13132x^2 + 13068x - 5040. \end{aligned}$$

解 $q(x)=x^6, \varepsilon=-0.002$.

$p(x)$的根 $x_k=k(k=1,\cdots,7)$, $p'(x_k)=\prod_{j\neq k}(k-j)$, $q(x_k)=k^6$,由式(5.14.3)可得

$$x_k(\varepsilon)\approx k+\frac{(-1)^{k-1}0.002k^6}{(k-1)!(7-k)!}=k+\delta(k).$$

$\delta(k)$的数值如下:

$$\delta(1)=2.78\text{E}-6,\quad \delta(2)=-1.07\text{E}-3,$$
$$\delta(3)=3.04\text{E}-2,\quad \delta(4)=-2.28\text{E}-1,$$
$$\delta(5)=6.51\text{E}-1,\quad \delta(6)=-7.77\text{E}-1,$$
$$\delta(7)=3.21\text{E}-1.$$

实际上,方程 $p(x)+\varepsilon x^6=0$ 的根 $x_k(\varepsilon)$分别为

1.0000028, 1.9989382, 3.0331253, 3.8195692

5.4586758±0.54012578i, 7.2330128.

这说明当 x^6 的系数相对误差是$-\frac{0.002}{28}=7.1\text{E}-5$时,根的误差很大,即系数微小变化引起根的不稳定.它说明对病态条件的多项式求根是困难的.由于用计算机计算时,系数在10→2转换中就可能产生微小误差,从而可能使算出的根不可信.为解决这个问题可采用双字长运算,另外,求根时先求量小的根,逐步算出大根,其效果较好.但是,如用降价方法求根则稳定性更差,此时可把得到的根作为近似值,再用牛顿法精确化,从而求得精度较高的根.

6 解线性方程组的直接方法

6.1 引言

大量的科学和工程计算问题归结为求解线性方程组

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \tag{6.1.1}$$

其中 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$，$\boldsymbol{b}\in\mathbb{R}^{n}$，$\boldsymbol{A}$ 是非奇异的.

所谓**直接法**是指假定在计算中没有舍入误差，经过有限次算术运算能给出 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 的精确解的数值方法. 对中小型问题，这类方法很有效.

直接法多数基于很容易求解的三角形方程组

$$\boldsymbol{L}\boldsymbol{x} = \boldsymbol{b}, \tag{6.1.2}$$

其中 $\boldsymbol{L}$ 为下三角阵($l_{ij}=0, i<j$)或

$$\boldsymbol{U}\boldsymbol{x} = \boldsymbol{b}, \tag{6.1.3}$$

其中 U 为上三角阵($u_{ij}=0, i>j$). 对于式(6.1.2)，当 $l_{ii}\neq 0(i=1,2,\cdots,n)$时，向前代入求得逐个分量，即

$$x_1 = b_1/l_{11},\quad x_i = \Big(b_i - \sum_{j=1}^{i-1} l_{ij}x_j\Big)/l_{ii},\quad i = 2,\cdots,n.$$

对于式(6.1.3)，当 $u_{ii}\neq 0(i=1,2,\cdots,n)$时，向后回代求得逐个分量，即

$$x_n = b_n/u_{nn},\quad x_i = \Big(b_i - \sum_{j=i+1}^{n} u_{ij}x_j\Big)/u_{ii},\quad i = n-1,\cdots,1.$$

直接法的基本思想是将 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 化成与之等价的上三角(或下三角)形式求解.

6.2 矩阵分析

6.2.1 向量范数

定义 6.2.1 如果对$\mathbb{R}^n$上任一向量$\boldsymbol{x}$,对应一个实数$\|\boldsymbol{x}\|$,它满足如下条件:

(1)(正定性)$\|\boldsymbol{x}\| \geqslant 0, \forall \boldsymbol{x} \in \mathbb{R}^n$,当且仅当$\boldsymbol{x}=\boldsymbol{0}$时,$\|\boldsymbol{x}\|=0$;

(2)(齐次性)$\|\alpha \boldsymbol{x}\| = |\alpha| \, \|\boldsymbol{x}\|, \alpha \in \mathbb{R}, \forall \boldsymbol{x} \in \mathbb{R}^n$;

(3)(三角不等式)$\|\boldsymbol{x}+\boldsymbol{y}\| \leqslant \|\boldsymbol{x}\| + \|\boldsymbol{y}\|, \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$.

则称$\|\boldsymbol{x}\|$为$\mathbb{R}^n$上$\boldsymbol{x}$的**范数**(norm).

$\mathbb{C}^n$中向量范数可以类似定义.在$\mathbb{R}^n$和$\mathbb{C}^n$中,设$\boldsymbol{x}=(x_1, x_2, \cdots, x_n)^{\mathrm{T}}$,一类有用的向量范数是$p$-范数,即

$$\|\boldsymbol{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{\frac{1}{p}}, \quad p \geqslant 1,$$

其中,最常用的是$p=1,2$和∞.

$$\|\boldsymbol{x}\|_1 = |x_1| + |x_2| + \cdots + |x_n|;$$

$$\|\boldsymbol{x}\|_2 = (|x_1|^2 + |x_2|^2 + \cdots + |x_n|^2)^{\frac{1}{2}} = (\boldsymbol{x}, \boldsymbol{x})^{\frac{1}{2}};$$

$$\|\boldsymbol{x}\|_\infty = \max_{1 \leqslant i \leqslant n} |x_i|.$$

它们分别称为1-范数,2-范数和∞-范数.

p-范数的一个经典结果为赫尔德(Hölder)不等式

$$|\boldsymbol{x}^{\mathrm{T}} \boldsymbol{y}| \leqslant \|\boldsymbol{x}\|_p \|\boldsymbol{y}\|_p, \quad \frac{1}{p} + \frac{1}{q} = 1.$$

它的一个非常重要的特殊情况是柯西-施瓦兹(Cauchy-Schwarz)不等式

$$|\boldsymbol{x}^{\mathrm{T}} \boldsymbol{y}| \leqslant \|\boldsymbol{x}\|_2 \|\boldsymbol{y}\|_2.$$

定理 6.2.2 向量空间$\mathbb{R}^n$上的所有范数都是彼此等价的.

定理说明:如果$\|\cdot\|_\alpha$和$\|\cdot\|_\beta$是$\mathbb{R}^n$上的范数,则存在正常

数 c_1 和 c_2，使

$$c_1 \| \boldsymbol{x} \|_\alpha \leqslant \| \boldsymbol{x} \|_\beta \leqslant c_2 \| \boldsymbol{x} \|_\alpha$$

对一切 $\boldsymbol{x} \in \mathbb{R}^n$ 都成立.

例如，$\boldsymbol{x} \in \mathbb{R}^n$，则有

$$\| \boldsymbol{x} \|_2 \leqslant \| \boldsymbol{x} \|_1 \leqslant \sqrt{n} \| \boldsymbol{x} \|_2;$$

$$\| \boldsymbol{x} \|_\infty \leqslant \| \boldsymbol{x} \|_2 \leqslant \sqrt{n} \| \boldsymbol{x} \|_\infty;$$

$$\| \boldsymbol{x} \|_\infty \leqslant \| \boldsymbol{x} \|_1 \leqslant n \| \boldsymbol{x} \|_\infty.$$

6.2.2 矩阵范数

定义 6.2.3 如果对 $\mathbb{R}^{m\times n}$ 上任一矩阵 $\boldsymbol{A}$，对应一个实数 $\| \boldsymbol{A} \|$，满足以下条件：

(1) $\| \boldsymbol{A} \| \geqslant 0, \boldsymbol{A} \in \mathbb{R}^{m\times n}$，当且仅当 $\boldsymbol{A}=\boldsymbol{0}$ 时，$\| \boldsymbol{A} \| = 0$；

(2) $\| \alpha \boldsymbol{A} \| = |\alpha| \| \boldsymbol{A} \|, \alpha \in \mathbb{R}, \boldsymbol{A} \in \mathbb{R}^{m\times n}$；

(3) $\| \boldsymbol{A}+\boldsymbol{B} \| \leqslant \| \boldsymbol{A} \| + \| \boldsymbol{B} \|, \boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{m\times n}$；

(4) $\| \boldsymbol{AB} \| \leqslant \| \boldsymbol{A} \| \| \boldsymbol{B} \|, \boldsymbol{A} \in \mathbb{R}^{m\times n}, \boldsymbol{B} \in \mathbb{R}^{n\times l}$.

则称 $\| \boldsymbol{A} \|$ 为矩阵 $\boldsymbol{A}$ 的范数，简称**矩阵范数**(matrix norm)

复矩阵 $\boldsymbol{A} \in \mathbb{C}^{m\times n}$ 范数的定义是完全类似的.

常用的矩阵范数有

(1) 佛罗比尼乌斯(Frobenius)范数

$$\| \boldsymbol{A} \|_{\mathrm{F}} = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} | a_{ij} |^2 \right)^{\frac{1}{2}}, \quad \boldsymbol{A} \in \mathbb{R}^{m\times n},$$

简称 $\boldsymbol{F}$-范数.

(2) 诱导矩阵范数

$$\| \boldsymbol{A} \|_p = \max_{\substack{x\neq 0 \\ x \in \mathbb{R}^n}} \frac{\| \boldsymbol{Ax} \|_p}{\| \boldsymbol{x} \|_p} = \max_{\| x \|_p = 1} \| \boldsymbol{Ax} \|_p, \quad \forall \boldsymbol{A} \in \mathbb{R}^{m\times n}.$$

称为已给向量范数的**从属范数**(subordinate norm)，也称为该向量诱导(induce)的矩阵范数，有时也称为 $\mathbb{R}^{n\times n}$ 上的**算子范数**

(operator norm),简称 p-范数.

p-范数的一个重要性质是:

$$\|Ax\|_p \leqslant \|A\|_p \|x\|_p, \quad \forall A \in \mathbb{R}^{m\times n}, \quad x \in \mathbb{R}^n.$$

定理 6.2.4 设 $A \in \mathbb{R}^{m\times n}$,则

$$\|A\|_\infty = \max_{1\leqslant i\leqslant m} \sum_{j=1}^{n} |a_{ij}|;$$

$$\|A\|_1 = \max_{1\leqslant j\leqslant n} \sum_{i=1}^{m} |a_{ij}|;$$

$$\|A\|_2 = (\lambda_{\max}(A^{\mathrm{T}}A))^{\frac{1}{2}}.$$

其中 $\lambda_{\max}(A^{\mathrm{T}}A)$ 表示对称半正定矩阵 $A^{\mathrm{T}}A$ 的最大特征值. 它也可以推广到 $\mathbb{C}^{m\times n}$,其中 $\|A\|_2 = (\lambda_{\max}(A^{\mathrm{H}}A))^{\frac{1}{2}}$.

$\mathbb{R}^{m\times n}$ 上的 F-范数和 p-范数(特别是 $p=1,2,\infty$)有如下等价关系:

$$\|A\|_2 \leqslant \|A\|_F \leqslant \sqrt{n}\|A\|_2;$$

$$\frac{1}{\sqrt{n}}\|A\|_\infty \leqslant \|A\|_2 \leqslant \sqrt{m}\|A\|_\infty;$$

$$\frac{1}{\sqrt{m}}\|A\|_1 \leqslant \|A\|_2 \leqslant \sqrt{n}\|A\|_1.$$

且有如下的不等式:

$$\max_{i,j}|a_{ij}| \leqslant \|A\|_2 \leqslant \sqrt{mn}\max_{i,j}|a_{ij}|;$$

$$\|A\|_2 \leqslant \sqrt{\|A\|_1 \|A\|_\infty};$$

$$\|A(i_1 : i_2, j_1 : j_2)\|_p \leqslant \|A\|_p.$$

其中 $1 \leqslant i_1 \leqslant i_2 \leqslant m, 1 \leqslant j_1 \leqslant j_2 \leqslant n$.

定理 6.2.5 设 $A \in \mathbb{C}^{m\times n}$,则

$$\|A\|_2 = \max\{|y^{\mathrm{H}}Ax| \mid x \in \mathbb{C}^n, \|x\|_2 = 1, y \in \mathbb{C}^m, \|y\|_2 = 1\};$$

$$\| A^H \|_2 = \| A^T \|_2 = \| A \|_2;$$
$$\| A^H A \|_2 = \| A \|_2^2.$$
其中 $A^H = \overline{A}^T$ 即矩阵 A 的共轭转置.

定理 6.2.6 设 $A \in \mathbb{C}^{n\times n}$,则对任意酉矩阵 $U \in \mathbb{C}^{n\times n}$ 和酉矩阵 $V \in \mathbb{C}^{n\times n}$,成立

$$\| UAV \|_2 = \| A \|_2,$$
$$\| UAV \|_F = \| A \|_F,$$

其中满足 $X^H X = I$ 的矩阵 X 称为酉矩阵(unitary matrix).

该定理说明矩阵的 2-范数和 F-范数是酉不变的.

显然,对于实矩阵 $A \in \mathbb{R}^{n\times n}$,以上的定义、定理和性质都是成立的.

定义 6.2.7 设 $A \in \mathbb{R}^{n\times n}$,$\lambda_i (i=1,2,\cdots,n)$ 为其特征值,则称 $\rho(A) = \max\limits_{1\leqslant i\leqslant n} |\lambda_i|$ 为矩阵 A 的**谱半径**(spectral radius).

定理 6.2.8 (1) 设 $\| \cdot \|$ 为 $\mathbb{R}^{n\times n}$ 上的任一种矩阵范数,则对任意 $A \in \mathbb{R}^{n\times n}$ 有 $\rho(A) \leqslant \| A \|$;

(2) 如果 $A \in \mathbb{R}^{n\times n}$ 为对称矩阵,则 $\rho(A) = \| A \|_2$;

(3) 对任意的 $A \in \mathbb{R}^{n\times n}$ 及实数 $\varepsilon > 0$,至少存在一种从属的矩阵范数 $\| \cdot \|_\varepsilon$,使 $\| A \|_\varepsilon \leqslant \rho(A) + \varepsilon$.

定理 6.2.9 设 $\| \cdot \|$ 是 $\mathbb{R}^{n\times n}$ 上的一种从属范数,矩阵 $B \in \mathbb{R}^{n\times n}$ 满足 $\| B \| < 1$,则

(1) $I + B$ 非奇异;

(2) $\| (I+B)^{-1} \| \leqslant \dfrac{1}{1 - \| B \|}$.

6.2.3 初等矩阵

定义 6.2.10 设 $u, v \in \mathbb{C}^n$,$\sigma \in \mathbb{C}$,称

$$E(u, v; \sigma) = I - \sigma u v^H$$

为**初等矩阵**(elementary matrices),其中 v^H 为 v 的共轭转置,即 $v^H = \overline{v}^T$. 当 $u, v \in \mathbb{R}^n$,$\sigma \in \mathbb{R}$ 时,称 $E(u, v; \sigma)$ 为**实初等矩阵**.

例如，若 $\boldsymbol{u}=(u_1,u_2,u_3)^{\mathrm{T}}, \boldsymbol{v}=(v_1,v_2,v_3)^{\mathrm{T}}\in\mathbb{R}^3, \sigma\in\mathbb{R}$ 时，我们有

$$\boldsymbol{E}(\boldsymbol{u},\boldsymbol{v};\sigma)=\begin{pmatrix}1-\sigma u_1v_1 & -\sigma u_1v_2 & -\sigma u_1v_3\\ -\sigma u_2v_1 & 1-\sigma u_2v_2 & -\sigma u_2v_3\\ -\sigma u_3v_1 & -\sigma u_3v_2 & 1-\sigma u_3v_3\end{pmatrix}.$$

定理 6.2.11 初等矩阵有性质：

(1) $\det\boldsymbol{E}(\boldsymbol{u},\boldsymbol{v};\sigma)=1-\sigma\boldsymbol{v}^{\mathrm{H}}\boldsymbol{u}$；

(2) 若 $1-\sigma\boldsymbol{v}^{\mathrm{H}}\boldsymbol{u}\neq0$，则初等矩阵 $\boldsymbol{E}(\boldsymbol{u},\boldsymbol{v};\sigma)$ 可逆，其逆也是初等矩阵：

$$\boldsymbol{E}(\boldsymbol{u},\boldsymbol{v};\sigma)^{-1}=\boldsymbol{E}(\boldsymbol{u},\boldsymbol{v};\tau),\quad\forall\sigma,\tau\in\mathbb{C}\quad 且\quad\tau=\frac{\sigma}{\sigma\boldsymbol{v}^{\mathrm{T}}\boldsymbol{u}-1}.$$

例 6.2.12 初等下三角矩阵.

设 $\sigma=-1, \boldsymbol{v}=\boldsymbol{e}_j, \boldsymbol{u}=\boldsymbol{l}_j=(0,\cdots,0,l_{j+1j},\cdots,l_{nj})^{\mathrm{T}}$，则 $\boldsymbol{E}(\boldsymbol{l}_j,\boldsymbol{e}_j;-1)$，记 $\boldsymbol{L}_j=\boldsymbol{L}_j(\boldsymbol{l}_j)=\boldsymbol{I}+\boldsymbol{l}_j\boldsymbol{e}_j^{\mathrm{T}}$，即

$$\boldsymbol{L}_j(\boldsymbol{l}_j)=\begin{pmatrix}1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & l_{j+1,j} & 1 & \\ & & \vdots & & \ddots & \\ & & l_{n,j} & & & 1\end{pmatrix}$$

称为**初等下三角矩阵**，也称为**初等消元矩阵**(elementary elimination matrices). 在线性代数方程组的高斯消元法中有重要作用.

初等下三角矩阵有如下性质：

(1) $\boldsymbol{L}_j^{-1}(\boldsymbol{l}_j)=\boldsymbol{L}_j(-\boldsymbol{l}_j)$；

(2) $\det\boldsymbol{L}_j(\boldsymbol{l}_j)=1$；

(3) 当 $i\leqslant j$ 时，有正交性 $\boldsymbol{e}_i^{\mathrm{T}}\boldsymbol{l}_j=0$，所以

$$\boldsymbol{L}=\boldsymbol{L}_1(\boldsymbol{l}_1)\boldsymbol{L}_2(\boldsymbol{l}_2)\cdots\boldsymbol{L}_{n-1}(\boldsymbol{l}_{n-1})=\boldsymbol{I}+\boldsymbol{l}_1\boldsymbol{e}_1^{\mathrm{T}}+\cdots+\boldsymbol{l}_{n-1}\boldsymbol{e}_{n-1}^{\mathrm{T}}$$

为单位下三角矩阵：

$$L=\begin{bmatrix}1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ l_{n1} & l_{n2} & \cdots & l_{n,n-1} & 1\end{bmatrix}.$$

(4) $\boldsymbol{L}_i$ 左乘 $\boldsymbol{A}$ 的结果是从 $\boldsymbol{A}$ 的各行(第 $i+1$ 行到第 n 行)分别加上 $\boldsymbol{A}$ 的第 i 行乘 1 个因子.

例 6.2.13 初等排列矩阵(elementary permutation matrices).

令 $\sigma=1,\boldsymbol{u}=\boldsymbol{v}=\boldsymbol{e}_i-\boldsymbol{e}_j$，则

$$\boldsymbol{P}_{ij}=\boldsymbol{E}(\boldsymbol{e}_i-\boldsymbol{e}_j,\boldsymbol{e}_i-\boldsymbol{e}_j;1)=\boldsymbol{I}-(\boldsymbol{e}_i-\boldsymbol{e}_j)(\boldsymbol{e}_i-\boldsymbol{e}_j)^{\mathrm{T}}$$

称为**初等排列矩阵**，也称为**初等置换矩阵**.

显然，$\boldsymbol{P}_{ij}$ 是由单位矩阵的第 i,j 行交换得到的，有如下性质：

(1) $\boldsymbol{P}_{ij}$ 是对称正交矩阵，即 $\boldsymbol{P}_{ij}^{\mathrm{T}}=\boldsymbol{P}_{ij},\boldsymbol{P}_{ij}^{\mathrm{T}}\boldsymbol{P}_{ij}=\boldsymbol{I}$；

(2) $\det \boldsymbol{P}_{ij}=-1$；

(3) $\boldsymbol{P}_{ij}\boldsymbol{A}$ 是将 $\boldsymbol{A}$ 的第 i,j 行交换所得到的矩阵，$\boldsymbol{A}\boldsymbol{P}_{ij}$ 是将 $\boldsymbol{A}$ 的第 i,j 列交换；

(4) 若干个初等排列阵的乘积仍称为排列矩阵，它相当于单位矩阵的各列交换了若干次.

例 6.2.14 初等埃尔米特矩阵.

令 $\sigma=2,\boldsymbol{u}=\boldsymbol{v}=\boldsymbol{w}\in\mathbb{C}^n$，且 $\|\boldsymbol{w}\|_2=1$，则

$$\boldsymbol{H}(\boldsymbol{w})=\boldsymbol{E}(\boldsymbol{w},\boldsymbol{w};2)=\boldsymbol{I}-2\boldsymbol{w}\boldsymbol{w}^{\mathrm{H}}$$

称为**初等埃尔米特矩阵**，即 $\boldsymbol{H}(\boldsymbol{w})^{\mathrm{H}}=\boldsymbol{H}(\boldsymbol{w})$. 它所表示的变换称为**豪斯霍尔德**(Householder)**变换**.

$\boldsymbol{H}(\boldsymbol{w})$是酉矩阵，即 $\boldsymbol{H}(\boldsymbol{w})^{\mathrm{H}}\boldsymbol{H}(\boldsymbol{w})=\boldsymbol{I}$，且 $\det\boldsymbol{H}(\boldsymbol{w})=-1$. 它在特征值问题计算中是最重要的工具之一(见 8.2.2 节).

6.3 高斯顺序消去法和矩阵的 LU 分解

设方程组

$$\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}, \tag{6.3.1}$$

其中 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 非奇异($\det\boldsymbol{A}\neq 0$),$\boldsymbol{b}\in\mathbb{R}^{n}$.

1. 高斯顺序消去法(sequential elimination)

高斯消去法就是将系数矩阵 $\boldsymbol{A}$ 逐次消元成上三角矩阵,再逐次向后回代,求解上三角形方程组,得到方程组(6.3.1)的解.

下面用初等消元矩阵(见例 6.2.12),逐次左乘式(6.3.1)的两边,完成高斯消元过程.

设增广矩阵$(\boldsymbol{A}^{(1)}\mid\boldsymbol{b}^{(1)})=(\boldsymbol{A}\mid\boldsymbol{b})$. 令 $a_{11}^{(1)}\neq 0$,$\boldsymbol{l}_1=(0,l_{21},\cdots,l_{n1})^{\mathrm{T}}$,其中 $l_{i1}=\dfrac{a_{i1}^{(1)}}{a_{11}^{(1)}}(i=2,\cdots,n)$为消元因子,则有初等消元矩阵

$$\boldsymbol{L}_1(-\boldsymbol{l}_1)=\boldsymbol{I}-\boldsymbol{l}_1\boldsymbol{e}_1^{\mathrm{T}}=\begin{pmatrix}1 & & & \\ -l_{21} & 1 & & \\ \vdots & & \ddots & \\ -l_{n1} & & & 1\end{pmatrix}.$$

第 1 次消元:用 $\boldsymbol{L}_1(-\boldsymbol{l}_1)$左乘$(\boldsymbol{A}^{(1)}\mid\boldsymbol{b}^{(1)})$,得

$$\begin{aligned}(\boldsymbol{A}^{(2)}\mid\boldsymbol{b}^{(2)})&=\boldsymbol{L}_1(-\boldsymbol{l}_1)(\boldsymbol{A}^{(1)}\mid\boldsymbol{b}^{(1)})\\&=\left(\begin{array}{cccc|c}a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)}\\ & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)}\\ & \vdots & & \vdots & \vdots\\ & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} & b_n^{(2)}\end{array}\right).\end{aligned}$$

其第 1 行元素未变,第 1 列对角元之下的元素已消成 0,$a_{ij}^{(2)}=a_{ij}^{(1)}-l_{i1}a_{1j}^{(1)}(i,j=2,3,\cdots,n)$,$b_i^{(2)}=b_i^{(1)}-l_{i1}b_1^{(1)}(i=2,3,\cdots,n)$.

经 $k-1$ 次消元后,有

$$
(\boldsymbol{A}^{(k)} \mid \boldsymbol{b}^{(k)}) = \left[\begin{array}{cccccc:c}
a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1k}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\
 & a_{22}^{(2)} & \cdots & a_{2k}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\
 & & \ddots & \vdots & & \vdots & \vdots \\
 & & & a_{kk}^{(k)} & \cdots & a_{kn}^{(k)} & b_k^{(k)} \\
 & & & \vdots & & \vdots & \vdots \\
 & & & a_{nk}^{(k)} & \cdots & a_{nn}^{(k)} & b_n^{(k)}
\end{array}\right]. \tag{6.3.2}
$$

设主元素(principal element)$a_{kk}^{(k)} \neq 0$，$\boldsymbol{l}_k = (0, \cdots, 0, l_{k+1,k}, \cdots, l_{nk})^{\mathrm{T}}$，其中 $l_{ik} = \dfrac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$ $(i = k+1, \cdots, n)$ 为第 k 次消元因子.

第 k 次消元：用 $\boldsymbol{L}_k(-\boldsymbol{l}_k) = \boldsymbol{I} - \boldsymbol{l}_k \boldsymbol{e}_k^{\mathrm{T}}$ 左乘$(\boldsymbol{A}^{(k)} \mid \boldsymbol{b}^{(k)})$，得

$$
(\boldsymbol{A}^{(k+1)} \mid \boldsymbol{b}^{(k+1)}) = \boldsymbol{L}_k(-\boldsymbol{l}_k)(\boldsymbol{A}^{(k)} \mid \boldsymbol{b}^{(k)})
$$

$$
= \left[\begin{array}{ccccccc:c}
a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1k}^{(1)} & a_{1,k+1}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\
 & a_{22}^{(2)} & \cdots & a_{2k}^{(2)} & a_{2,k+1}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\
 & & \ddots & \vdots & \vdots & & \vdots & \vdots \\
 & & & a_{kk}^{(k)} & a_{k,k+1}^{(k)} & \cdots & a_{kn}^{(k)} & b_k^{(k)} \\
 & & & & a_{k+1,k+1}^{(k+1)} & \cdots & a_{k+1,n}^{(k+1)} & b_{k+1}^{(k+1)} \\
 & & & & \vdots & & \vdots & \vdots \\
 & & & & a_{n,k+1}^{(k+1)} & \cdots & a_{nn}^{(k+1)} & b_n^{(k+1)}
\end{array}\right],
$$

其中 $a_{ij}^{(k+1)} = a_{ij}^{(k)} - l_{ik} a_{kj}^{(k)}$，$i, j = k+1, \cdots, n$；$b_i^{(k+1)} = b_i^{(k)} - l_{ik} b_k^{(k)}$，$i = k+1, \cdots, n$.

仿上述过程，直到 $n-1$ 步，可完成高斯消元过程，得三角形方程组.

$$
\boldsymbol{A}^{(n)} \boldsymbol{x} = \boldsymbol{b}^{(n)},
$$

其中 $\boldsymbol{A}^{(n)}$ 为上三角矩阵，其对角元 $a_{kk}^{(k)}$ $(k = 1, 2, \cdots, n)$ 均为主元素.

对上式用回代过程，求得原方程组(6.3.1)的解如下：

$$
\begin{cases}
x_n = b_n^{(n)} / a_{nn}^{(n)} \\
x_k = \left(b_k^{(k)} - \displaystyle\sum_{j=k+1}^{n} a_{kj}^{(k)} x_j \right) \Big/ a_{kk}^{(k)}, \quad k = n-1, \cdots, 2, 1.
\end{cases}
$$

高斯消元过程和回代过程合起来称为顺序高斯消去法.

高斯消去法的乘除法次数为$\frac{n^3}{3}+n^2-\frac{n}{3}\approx\frac{n^3}{3}$,加减法次数为$\frac{n^3}{3}+\frac{n^2}{2}-\frac{5}{6}n\approx\frac{n^3}{3}$.

2. LU 分解

由初等消元矩阵(见例 6.2.12)的性质:$\boldsymbol{L}_j^{-1}(\boldsymbol{l}_j)=\boldsymbol{L}_j(-\boldsymbol{l}_j)$,综合 $n-1$ 步消元过程,有

$$\boldsymbol{L}_{n-1}(-\boldsymbol{l}_{n-1})\boldsymbol{L}_{n-2}(-\boldsymbol{l}_{n-2})\cdots\boldsymbol{L}_1(-\boldsymbol{l}_1)\boldsymbol{A}=\boldsymbol{A}^{(n)}.$$

它完成了对矩阵 $\boldsymbol{A}$ 的 LU 分解,即

$$\boldsymbol{A}=\boldsymbol{L}\boldsymbol{U},$$

其中 $\boldsymbol{L}$ 为单位下三角矩阵,$\boldsymbol{U}$ 为上三角矩阵,即

$$\begin{aligned}\boldsymbol{L}&=\boldsymbol{L}_1(\boldsymbol{l}_1)\boldsymbol{L}_2(\boldsymbol{l}_2)\cdots\boldsymbol{L}_{n-1}(\boldsymbol{l}_{n-1})\\&=\boldsymbol{I}+\boldsymbol{l}_1\boldsymbol{e}_1^{\mathrm{T}}+\cdots+\boldsymbol{l}_{n-1}\boldsymbol{e}_{n-1}^{\mathrm{T}}\\&=\begin{pmatrix}1&&&&\\l_{21}&1&&&\\l_{31}&l_{32}&1&&\\\vdots&\vdots&\ddots&\ddots&\\l_{n1}&l_{n2}&\cdots&l_{n,n-1}&1\end{pmatrix},\end{aligned}$$

$$\boldsymbol{U}=\boldsymbol{A}^{(n)}=\begin{pmatrix}a_{11}^{(1)}&a_{12}^{(1)}&\cdots&a_{1n}^{(1)}\\&a_{22}^{(2)}&\cdots&a_{2n}^{(2)}\\&&\ddots&\vdots\\&&&a_{nn}^{(n)}\end{pmatrix}.$$

$\boldsymbol{A}$ 的 LU 分解也称为 Doolittle 分解. 这时求解方程组(6.3.1)等于顺序求解如下的两个三角形方程组

$$\boldsymbol{L}\boldsymbol{y}=\boldsymbol{b},$$
$$\boldsymbol{U}\boldsymbol{x}=\boldsymbol{y},$$

它与高斯消去法是等价的.

高斯消元过程或 A 的 LU 分解能顺利进行的条件是主元素 $a_{kk}^{(k)} \neq 0(k=1,2,\cdots,n-1)$，回代步骤要求 $a_{nn}^{(n)} \neq 0$.

若用 D_k 表示 A 的顺序主子式，即主子矩阵 A_k 的行列式

$$D_k = \begin{vmatrix} a_{11} & \cdots & a_{1k} \\ a_{21} & \cdots & a_{2k} \\ \vdots & & \vdots \\ a_{k1} & \cdots & a_{kk} \end{vmatrix}, \quad k = 1,2,\cdots,n,$$

则有如下定理.

定理 6.3.1 $a_{kk}^{(k)}(k=1,2,\cdots,m)$ 均不为零的充分必要条件是 A 的顺序主子式 $D_k \neq 0, k=1,2,\cdots,m$，其中 $m \leqslant n$.

定理 6.3.2 若 $D_k \neq 0(k=1,2,\cdots,n)$，则可用高斯顺序消去法，求出方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 的解.

定理 6.3.3 若 $D_k \neq 0(k=1,2,\cdots,n-1)$，则 A 可分解为一个单位下三角矩阵 L 和一个上三角矩阵 U 的乘积，且这种分解是惟一的.

推论 6.3.4 设 $a_{ii}^{(i)} \neq 0(i=1,2,\cdots,k)$，则 $\det A_k = \prod_{i=1}^{k} a_{ii}^{(i)}, k=1,2,\cdots,n$.

易证明当 A 为对称正定矩阵或严格对角占优矩阵 $\left(|a_{ii}| > \sum_{\substack{j=1\\j\neq i}}^{n} |a_{ij}|, i=1,2,\cdots,n\right)$ 或不可约弱对角占优矩阵 $\left(|a_{ii}| \geqslant \sum_{\substack{j=1\\j\neq i}}^{n} |a_{ij}|, i=1,2,\cdots,n\right.$，且至少有一个不等式成立$\Big)$ 时，均满足定理 6.3.2 和定理 6.3.3 的条件，不必换行. 顺序完成高斯消元过程及对 A 作 LU 分解.

6.4 高斯主元素消去法

顺序高斯消去法在消元过程中可能出现零主元，即 $a_{kk}^{(k)}=0$ 的情况，这时计算出现溢出，消去法将无法进行；也可能 $a_{kk}^{(k)} \neq 0$，但

其绝对值非常小，也会由于用它做除法导致舍人误差的严重增长和扩散，使数值解不可靠.

主元素消去法是对顺序消去法的改进，它在消元过程的每一步局部地或全局地选取系数矩阵（或消元后的右下部分低阶子矩阵）中绝对值最大的元素为主元素，以使高斯消去法有较好的数值稳定性.

如果对非奇异阵 $\boldsymbol{A}$ 消元的第 k 步，从 $\boldsymbol{A}^{(k)}$（见式(6.3.2)）的第 k 列对角线元素之下选

$$|a_{i_k,k}^{(k)}| = \max_{k\leqslant j\leqslant n} |a_{jk}^{(k)}|$$

作为主元. 若 $i_k \neq k$，交换增广矩阵（见式(6.3.2)）的第 i_k, k 行，把 $a_{i_k,k}^{(k)}$ 调到主元素 $a_{kk}^{(k)}$ 的位置上，使消元因子 $|l_{ik}| = \dfrac{|a_{ik}^{(k)}|}{|a_{kk}^{(k)}|} \leqslant 1$（$i = k+1,\cdots,n$）达到抑制舍人误差的作用，每进行一次高斯消元之前，按列选一次主元，直到 $n-1$ 步，称这种方法为**列主元或部分选主元**(partial pivoting).

记第 k 次初等置换阵为

$$\boldsymbol{P}_k = \boldsymbol{P}_{i_k,k} = \begin{bmatrix} 1 &&&&&&&&& \\ & \ddots &&&&&&&& \\ && 1 &&&&&&& \\ &&& 0 &&&& 1 &&& \\ &&&& 1 &&&&& \\ &&&&& \ddots &&&& \\ &&&&&& 1 &&& \\ &&& 1 &&&& 0 && \\ &&&&&&&& 1 & \\ &&&&&&&&& \ddots & \\ &&&&&&&&&& 1 \end{bmatrix} \begin{matrix} \\ \\ \\ k \\ \\ \\ \\ i_k. \\ \\ \\ \\ \end{matrix}$$

定理 6.4.1 （部分选主元三角分解）对非奇异矩阵 $\boldsymbol{A}$，存在排列矩阵 $\boldsymbol{P}$，使

$$\boldsymbol{PA} = \boldsymbol{LU},$$

其中 $\boldsymbol{P}=\boldsymbol{P}_{n-1}\boldsymbol{P}_{n-2}\cdots\boldsymbol{P}_2\boldsymbol{P}_1$，$\boldsymbol{L}$ 为其元素的绝对值不大于 1 的单位下三角矩阵，$\boldsymbol{U}$ 为非奇异上三角矩阵.

这时，求解方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 等价于求解如下的两个三角形方程组

$$\begin{cases}\boldsymbol{Ly}=\boldsymbol{Pb},\\ \boldsymbol{Ux}=\boldsymbol{y}.\end{cases}$$

例 6.4.2 设 $\boldsymbol{A}=\begin{pmatrix}1&2&2\\4&4&2\\4&6&4\end{pmatrix}$，用高斯消去法和列主元消去法分别对 $\boldsymbol{A}$ 作 LU 分解

解 $\boldsymbol{A}$ 的 LU 分解为

$$\boldsymbol{A}=\begin{pmatrix}1&&\\4&1&\\4&0.5&1\end{pmatrix}\begin{pmatrix}1&2&2\\0&-4&-6\\0&0&-1\end{pmatrix}=\boldsymbol{LU}.$$

列主元三角分解为

$$\boldsymbol{PA}=\begin{pmatrix}1&&\\1&1&\\0.25&0.5&1\end{pmatrix}\begin{pmatrix}4&4&2\\0&2&2\\0&0&0.5\end{pmatrix}=\boldsymbol{LU},$$

其中

$$\boldsymbol{P}=\boldsymbol{P}_2\boldsymbol{P}_1=\begin{pmatrix}1&0&0\\0&0&1\\0&1&0\end{pmatrix}\begin{pmatrix}0&1&0\\1&0&0\\0&0&1\end{pmatrix}=\begin{pmatrix}0&1&0\\0&0&1\\1&0&0\end{pmatrix}.$$

如果在消元的第 k 步，从 $\boldsymbol{A}^{(k)}$（见式(6.3.2)）的右下角子矩阵

$$\begin{pmatrix}a_{kk}^{(k)}&\cdots&a_{kn}^{(k)}\\\vdots&&\vdots\\a_{nk}^{(k)}&\cdots&a_{nn}^{(k)}\end{pmatrix}$$

中选绝对值最大的元素，即

$$|a_{i_k,j_k}^{(k)}|=\max_{k\leqslant i,j\leqslant n}|a_{ij}^{(k)}|$$

作为主元素，然后将$(A^{(k)}|b^{(k)})$的第i_k,k行交换，第j_k,k列交换，同时将自变量x的第j_k,k的位置交换，并记录自变量的排列次序，称这种方法为**完全选主元法**(complete pivoting).

定理 6.4.3 （全主元三角分解）对非奇异阵A，存在排列矩阵P和Q，使

$$PAQ = LU,$$

其中L为单位下三角矩阵，其元素的绝对值不超过1，U为上三角矩阵.

为求解方程组$Ax=b$，等价于首先求解如下的三角形方程组

$$\begin{cases} Ly = Pb, \\ Uz = y, \end{cases}$$

最后得

$$x = Qz.$$

完全选主元消去法比列主元消去法有更好的数值稳定性，但计算量大得多. 一般情况下部分选主元消去法的舍入误差是小的，因此，它是求解线性方程组最实用的方法之一. 有一类矩阵，用高斯消去法是不必选主元的，如对称正定矩阵或按列对角占优矩阵.

6.5 高斯-若尔当消去法

6.5.1 列主元高斯-若尔当消去法

高斯-若尔当(Gauss-Jordan)**消去法**是高斯消去法的一种修正，即消去矩阵对角线上方和下方的元素，将矩阵A约化为单位矩阵.

设高斯-若尔当消去法已完成$k-1$步，将方程组$Ax=b$约化为$A^{(k)}x=b^{(k)}$，其增广矩阵

$$
(\boldsymbol{A}^{(k)} \mid \boldsymbol{b}^{(k)}) = \left(\begin{array}{cccccccc:c}
1 & & & & a_{1k}^{(k)} & \cdots & a_{1n}^{(k)} & & b_1^{(k)} \\
& 1 & & & a_{2k}^{(k)} & \cdots & a_{2n}^{(k)} & & b_2^{(k)} \\
& & \ddots & & \vdots & & \vdots & & \vdots \\
& & & 1 & a_{k-1,k}^{(k)} & \cdots & a_{k-1,n}^{(k)} & & b_{k-1}^{(k)} \\
& & & & a_{k,k}^{(k)} & \cdots & a_{k,n}^{(k)} & & b_k^{(k)} \\
& & & & \vdots & & \vdots & & \vdots \\
& & & & a_{nk}^{(k)} & \cdots & a_{n,n}^{(k)} & & b_n^{(k)}
\end{array}\right)
$$

第 k 步消元时，用消元矩阵 $\boldsymbol{M}_k$ 左乘 $\boldsymbol{A}^{(k)}\boldsymbol{x}=\boldsymbol{b}^{(k)}$ 两边，则 $\boldsymbol{M}_k\boldsymbol{A}^{(k)}$ 的第 k 列为

$$
\boldsymbol{M}_k(\boldsymbol{A}^{(k)}\boldsymbol{e}_k^{\mathrm{T}}) = \begin{pmatrix}
1 & & & m_{1k} & & & \\
& \ddots & & \vdots & & & \\
& & 1 & m_{k-1,k} & & & \\
& & & \dfrac{1}{a_{kk}^{(k)}} & & & \\
& & & m_{k+1,k} & 1 & & \\
& & & \vdots & & \ddots & \\
& & & m_{n,k} & & & 1
\end{pmatrix}
\begin{pmatrix} a_{1k}^{(k)} \\ \vdots \\ a_{k-1,k}^{(k)} \\ a_{kk}^{(k)} \\ a_{k+1,k}^{(k)} \\ \vdots \\ a_{n,k}^{(k)} \end{pmatrix}
= \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \tag{6.5.1}
$$

其中 $m_{ik}=-\dfrac{a_{ik}^{(k)}}{a_{kk}^{(k)}}, i=1,\cdots,k-1,k+1,\cdots,n, m_k=\dfrac{1}{a_{kk}^{(k)}}$.

完成 $n-1$ 步消元后，有

$$
(\boldsymbol{A} \mid \boldsymbol{b}) = (\boldsymbol{A}^{(1)} \mid \boldsymbol{b}^{(1)}) \rightarrow (\boldsymbol{A}^{(n)} \mid \boldsymbol{b}^{(n)}) = (\boldsymbol{I} \mid \boldsymbol{b}^{(n)}),
$$

其中 $\boldsymbol{b}^{(n)}=(b_1^{(n)},b_2^{(n)},\cdots,b_n^{(n)})^{\mathrm{T}}$ 就是 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 的解 $\boldsymbol{x}$.

算法 6.5.1 列主元高斯-若尔当消去法

给定非奇异阵 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$，用列主元高斯-若尔当消去法求 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 的解.

(1) 对于 $k=1,2,\cdots,n$

(2) 选列主元即确定 i_k，使 $|a_{i_k,k}|=\max\limits_{k\leqslant i\leqslant n}|a_{ik}|$.

(3) 如果 $a_{i_k,k}=0$,则停止；否则

如果 $i_k=k$,则转(4)；

否则交换$(\boldsymbol{A}|\boldsymbol{b})$的第 i_k,k 行,即

$$a_{k_j}\leftrightarrow a_{i_k,j},\quad j=k,\quad k+1,\quad \cdots,\quad n$$

$$b_k\leftrightarrow b_{i_k}$$

(4) 计算乘数 $a_{ik}\leftarrow m_{ik}=-\dfrac{a_{ik}}{a_{kk}},\quad i=1,2,\cdots,n,\quad i\neq k$

$$a_{kk}\leftarrow m_{kk}=\frac{1}{a_{kk}}$$

(5) 消元计算

$$a_{ij}\leftarrow a_{ij}+m_{ik}a_{kj},\quad i=1,2,\cdots,n\quad 且\quad i\neq k,\quad j=k+1,\cdots,n$$

$$b_i\leftarrow b_i+m_{ik}b_k,\quad i=1,2,\cdots,n\quad 且\quad i\neq k$$

(6) 计算主行(第 k 行)

$$a_{kj}\leftarrow a_{kj}m_{kk},\quad j=k,k+1,\cdots,n$$

$$b_k\leftarrow b_km_{kk}.$$

高斯-若尔当消去法的总乘除法次数约$\dfrac{n^3}{2}+n^2-\dfrac{n}{2}$,比高斯消去法的计算量大.但用它求一个矩阵的逆矩阵还是比较合适的.

6.5.2 高斯-若尔当消去法求逆矩阵

设 $\boldsymbol{AX}=\boldsymbol{I}$,其中 $\boldsymbol{A},\boldsymbol{X},\boldsymbol{I}\in\mathbb{R}^{n\times n}$,$\boldsymbol{A}$ 非奇异,$\boldsymbol{I}$ 为单位矩阵,$\boldsymbol{X}=\boldsymbol{A}^{-1}$为 $\boldsymbol{A}$ 的逆矩阵.

将 $\boldsymbol{AX}=\boldsymbol{I}$ 改写为

$$\boldsymbol{A}(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n)=(\boldsymbol{e}_1,\boldsymbol{e}_2,\cdots,\boldsymbol{e}_n),$$

其中 $\boldsymbol{e}_i,\boldsymbol{x}_i\in\mathbb{R}^n,i=1,2,\cdots,n.$

求 $\boldsymbol{A}^{-1}$等价于用高斯-若尔当消去法同时求解如下 n 个方程组

$$\boldsymbol{Ax}_j=\boldsymbol{e}_j,\quad j=1,2,\cdots,n,$$

也等价于对增广矩阵 $\boldsymbol{C}=(\boldsymbol{A}|\boldsymbol{I})$应用高斯-若尔当消去法约化为

$(\boldsymbol{I}|\boldsymbol{B})$,则可得到 $\boldsymbol{A}^{-1}=\boldsymbol{B}$.

例 6.5.2 用列主元高斯-若尔当法求

$$\boldsymbol{A}=\begin{pmatrix}1 & 2 & 3\\2 & 4 & 5\\3 & 5 & 6\end{pmatrix}$$

的逆矩阵 $\boldsymbol{A}^{-1}$.

解 $$\boldsymbol{C}=\left(\begin{array}{ccc:ccc}1 & 2 & 3 & 1 & 0 & 0\\2 & 4 & 5 & 0 & 1 & 0\\3 & 5 & 6 & 0 & 0 & 1\end{array}\right)\xrightarrow[r_1\leftrightarrow r_3]{\text{换行}}\left(\begin{array}{ccc:ccc}3 & 5 & 6 & 0 & 0 & 1\\2 & 4 & 5 & 0 & 1 & 0\\1 & 2 & 3 & 1 & 0 & 0\end{array}\right)$$

$$\xrightarrow{\text{第 1 次消元}}\left(\begin{array}{ccc:ccc}1 & 5/3 & 2 & 0 & 0 & 1/3\\0 & 2/3 & 1 & 0 & 1 & -2/3\\0 & 1/3 & 1 & 1 & 0 & -1/3\end{array}\right)$$

$$\boldsymbol{c}_6$$

$$\xrightarrow{\text{第 2 次消元}}\left(\begin{array}{ccc:ccc}1 & 0 & -\frac{1}{2} & 0 & -5/2 & 2\\0 & 1 & 3/2 & 0 & 3/2 & -1\\0 & 0 & 1/2 & 1 & -1/2 & 0\end{array}\right)$$

$$\boldsymbol{c}_5$$

$$\xrightarrow{\text{第 3 次消元}}\left(\begin{array}{ccc:ccc}1 & 0 & 0 & 1 & -3 & 2\\0 & 1 & 0 & -3 & 3 & -1\\0 & 0 & 1 & 2 & -1 & 0\end{array}\right)=(\boldsymbol{I}\mid\boldsymbol{A}^{-1}).$$

$$\boldsymbol{c}_4$$

为节省内存空间,可不必存放数组 $\boldsymbol{C}$ 中的单位矩阵. 令式(6.5.1)中消元矩阵 $\boldsymbol{M}_k$ 的第 k 列为

$$\boldsymbol{m}_k=\left(-\frac{a_{1k}}{a_{kk}},\cdots,-\frac{1}{a_{kk}},\cdots,-\frac{a_{nk}}{a_{kk}}\right)^{\mathrm{T}},$$

则 $\boldsymbol{m}_1=\left(\frac{1}{3},-\frac{2}{3},-\frac{1}{3}\right)^{\mathrm{T}}=\boldsymbol{c}_6$, $\quad\boldsymbol{m}_2=\left(-\frac{5}{2},\frac{3}{2},-\frac{1}{2}\right)^{\mathrm{T}}=\boldsymbol{c}_5$,

$\boldsymbol{m}_3=(1,-3,2)^{\mathrm{T}}=\boldsymbol{c}_4$. 将 $\boldsymbol{c}_6$ 存放在 $\boldsymbol{c}_1$, $\boldsymbol{c}_5$ 存放在 $\boldsymbol{c}_2$, $\boldsymbol{c}_4$ 存放在 $\boldsymbol{c}_3$. 最后在 $\boldsymbol{C}$ 中 $\boldsymbol{A}$ 的位置上得到 $\boldsymbol{A}_1^{-1}$, 即 $\boldsymbol{A}_1^{-1}=(\boldsymbol{PA})^{-1}$. 从而有 $\boldsymbol{A}^{-1}=\boldsymbol{A}_1^{-1}\boldsymbol{P}$, 其中 $\boldsymbol{P}$ 为排列矩阵.

算法 6.5.3　列主元高斯-若尔当法求逆矩阵

求矩阵 $\boldsymbol{A}$ 的逆矩阵 $\boldsymbol{A}^{-1}$. 计算结果 $\boldsymbol{A}^{-1}$ 存放在原矩阵 $\boldsymbol{A}$ 的单元中. 用数组 $l(1:n)$ 记录主行, $\boldsymbol{A}$ 的行列式值存放在单元 det 中.

(1) det $:=1$

(2) 消元过程

对于 $k=1,2,\cdots,n$

① 选主元即确定 i_k, 使 $|a_{i_k,k}|=\max\limits_{k\leqslant i\leqslant n}|a_{ik}|$, $c:=a_{i_k k}$, $l(k):=i_k$

② 如果 $c=0$ 则 $\det\boldsymbol{A}=0$, 计算停止.

③ 否则如果 $i_k=k$ 则转(4).

否则换行 $a_{kj}\leftrightarrow a_{i_k j}$, $j=1,2,\cdots,n$

且 det $:=-$det

④ 计算 det $:=c$det

⑤ 计算主行元素

$$a_{kk}:=1;\quad a_{kj}:=a_{kj}/c,\quad j=1,2,\cdots,n$$

⑥ 约化非主行元素

对于 $i=1,2,\cdots,n,\quad i\neq k$

$w:=a_{ik};\quad a_{ik}:=0$

$a_{ij}:=a_{ij}-a_{kj}w,\quad j=1,2,\cdots,n$

(3) 调整列过程

对于 $k=n-1,n-2,\cdots,1$

① $m:=l(k)$

② 如果 $m\neq k$ 则换列

$$a_{ik}\longleftrightarrow a_{im},\ i=1,2,\cdots,n$$

6.6 直接三角分解法

6.3 节是用 $n-1$ 步高斯消元过程将矩阵 $\boldsymbol{A}$ 作了 LU 分解，即

$$\boldsymbol{A} = \boldsymbol{LU}. \tag{6.6.1}$$

直接三角分解法就是直接从公式(6.6.1)出发，通过矩阵乘法规则，建立计算三角形矩阵 $\boldsymbol{L}$ 和 $\boldsymbol{U}$ 的元素的递推公式.且对一些特殊类型的矩阵(如对称正定阵，三对角阵)的三角分解可以简化计算.

6.6.1 杜利特尔分解法

设矩阵 $\boldsymbol{A}$ 满足定理 6.3.3 的条件，则 $\boldsymbol{A}$ 可作 LU 分解(6.6.1)，其中 $\boldsymbol{L}$ 为单位下三角矩阵($l_{ii}=1, l_{ij}=0, i<j$)，$\boldsymbol{U}$ 为上三角矩阵($u_{ij}=0, i>j$).根据矩阵的乘法规则，有计算 $\boldsymbol{L}$ 和 $\boldsymbol{U}$ 元素的如下递推公式：

$$u_{kj} = a_{kj} - \sum_{r=1}^{k-1} l_{kr} u_{rj}, \quad j = k, k+1, \cdots, n, \tag{6.6.2}$$

$$l_{ik} = \left(a_{ik} - \sum_{r=1}^{k-1} l_{ir} u_{rk}\right) \Big/ u_{kk}, \quad i = k+1, \cdots, n. \tag{6.6.3}$$

对于 $k=1,2,\cdots,n$ 交替使用以上公式(6.6.2)和式(6.6.3)就能逐次计算出 $\boldsymbol{U}$(按行)和 $\boldsymbol{L}$(按列)的全部元素.且将它们存放在 $\boldsymbol{A}$ 的相应位置上($\boldsymbol{L}$ 的对角元 1 不存)，这就完成了 $\boldsymbol{A}$ 的 LU 分解.矩阵的分解式(6.6.2)和式(6.6.3)称为杜利特尔(Doolittle)分解.其中认为和式 $\sum\limits_{r=1}^{0}$ 为零.

求解 $\boldsymbol{Ly}=\boldsymbol{b}$ 和 $\boldsymbol{Ux}=\boldsymbol{y}$ 的计算公式为

$$\begin{cases} y_1 = b_1, \\ y_i = b_i - \sum\limits_{k=1}^{i-1} l_{ik} y_k, \quad i = 2, 3, \cdots, n; \end{cases}$$

$$\begin{cases} x_n = y_n / u_{nn}, \\ x_i = \left(y_i - \sum_{k=i+1}^{n} u_{ik} x_k \right) / u_{ii}, \quad i = n-1, \cdots, 1. \end{cases}$$

所得到的 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}$ 就是方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 的解.

直接三角分解法约需 $\frac{n^3}{3}$ 次乘除法，其计算量与高斯消去法基本相同.

6.6.2 列主元三角分解法

直接三角分解公式(6.6.3)中若出现 $u_{rr}=0$ 或 $|u_{rr}|$ 非常小，则杜利特尔分解会中断或引起大的舍入误差. 因此可采用与列主元消去法类似的方法，将直接三角分解法修改为列主元三角分解法.

由定理 6.4.1，对于非奇异矩阵 $\boldsymbol{A}$，则存在排列矩阵 $\boldsymbol{P}$，使得

$$\boldsymbol{PA} = \boldsymbol{LU},$$

其中 $\boldsymbol{L}$ 为单位下三角矩阵，$\boldsymbol{U}$ 为上三角矩阵. 这里我们在 $\boldsymbol{A}$ 的直接三角分解过程中采用选列主元，实现 $\boldsymbol{PA}$ 的 LU 分解. 再根据 $\boldsymbol{PAx}=\boldsymbol{Pb}$，通过求解 $\boldsymbol{Ly}=\boldsymbol{Pb}$ 和 $\boldsymbol{Ux}=\boldsymbol{y}$，得到原方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 的解.

令 $\boldsymbol{A}^{(1)}=\boldsymbol{A}$，设 $\boldsymbol{A}^{(1)}$ 的第 $k-1$ 步选列主元的分解已完成，且将已算出的 $\boldsymbol{L}$ 和 $\boldsymbol{U}$ 的元素存放在 $\boldsymbol{A}$ 的相应位置，记作

$$\boldsymbol{A}^{(k)} = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1,k-1} & u_{1k} & \cdots & u_{1n} \\ l_{21} & u_{22} & \cdots & u_{2,k-1} & u_{2k} & \cdots & u_{2n} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ l_{k-1,1} & l_{k-1,2} & \cdots & u_{k-1,k-1} & u_{k-1,k} & \cdots & u_{k-1,n} \\ l_{k,1} & l_{k2} & \cdots & l_{k,k-1} & a_{kk} & \cdots & a_{kn} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{n,k-1} & a_{nk} & \cdots & a_{nn} \end{bmatrix}.$$

此时，由于行交换，$A^{(k)}$ 中右下方的子阵内的 a_{ij} 可能已不是原来 A 中(i,j)位置上的元素.

第 k 步分解需利用形如式(6.6.2)和式(6.6.3)的公式，为避免用零或绝对值小的数做除法，引入量

$$s_i = a_{ik} - \sum_{r=1}^{k-1} l_{ir} u_{rk}, \quad i = k, k+1, \cdots, n,$$

即式(6.6.3)右端的分子部分 $s_{k+1}, \cdots, s_n$ 及增加了一个 s_k. 再在它们中间选取绝对值最大者，记作 s_{i_k}，即

$$|s_{i_k}| = \max_{k \leqslant i \leqslant n} |s_i|.$$

将 $s_k, s_{k+1}, \cdots, s_n$ 分别存放在 $A^{(k)}$ 的 $a_{kk}, a_{k+1,k}, \cdots, a_{nk}$ 的位置上，然后交换 $A^{(k)}$ 的第 k, i_k 行的元素. 将 s_{i_k} 调到了 $A^{(k)}$ 的(k,k)位置，但每个位置上的元素仍用原来的记号. 于是

$$u_{kk} = s_{i_k}, \quad s = s_{i_k},$$

$$u_{kj} = a_{kj} - \sum_{r=1}^{k-1} l_{kr} u_{rj}, \quad j = k+1, \cdots, n,$$

$$l_{ik} = s_i / s, \quad i = k+1, \cdots, n.$$

算法 6.6.1 列主元三角分解

用列主元三角分解法求解方程组 $\boldsymbol{Ax}=\boldsymbol{b}$，其中矩阵 $\boldsymbol{A}$ 非奇异. $\boldsymbol{PA}$ 的 LU 分解得到的 $\boldsymbol{L}$ 和 $\boldsymbol{U}$ 存放在 $\boldsymbol{A}$ 的相应位置，解 $\boldsymbol{x}$ 存放在 $\boldsymbol{b}$.

(1) $\boldsymbol{PA}$ 的 LU 分解过程

对于 $k=1,2,\cdots,n$

① 计算 s_i

$$a_{ik} \leftarrow s_i = a_{ik} - \sum_{r=1}^{k-1} l_{ir} u_{rk}, \quad i = k, k+1, \cdots, n.$$

② 选主元，即确定 i_k，使 $|s_{i_k}| = \max\limits_{k \leqslant i \leqslant n} |a_{ik}|$

③ 如果 $i_k = k$ 则转(4).

否则换行：$\begin{matrix} a_{kj} \leftrightarrow a_{i_k j}, & j=1,2,\cdots,n \\ b_k \leftrightarrow b_{i_k} & \end{matrix}$

④ 计算 $\boldsymbol{U}$ 的第 k 行元素

$$a_{kk} \leftarrow u_{kk} = s_{i_k}$$

$$a_{kj} \leftarrow u_{kj} = a_{kj} - \sum_{r=1}^{k-1} l_{kr} u_{rj}, \quad j = k+1, \cdots, n$$

⑤ 计算 $\boldsymbol{L}$ 的第 k 列元素

$$a_{ik} \leftarrow l_{ik} = a_{ik} / a_{kk}, \quad i = k+1, \cdots, n$$

(2) 方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 的求解过程(以上过程已在 $\boldsymbol{b}$ 的位置得到 $\boldsymbol{Pb}$)

① 求解 $\boldsymbol{Ly}=\boldsymbol{Pb}$

$$b_1 \leftarrow y_1 = b_1$$

$$b_i \leftarrow y_i = b_i - \sum_{j=1}^{i-1} l_{ij} b_j, \quad i = 2, 3, \cdots, n$$

② 求解 $\boldsymbol{Ux}=\boldsymbol{y}$($\boldsymbol{y}$ 存放在 $\boldsymbol{b}$ 中)

$$b_n \leftarrow x_n = b_n / u_{nn}$$

$$b_i \leftarrow x_i = \left(b_i - \sum_{j=i+1}^{n} u_{ij} x_j\right) / u_{ii}, \quad i = n-1, \cdots, 1$$

6.6.3 对称正定矩阵的楚列斯基分解法

设方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 的系数矩阵 $\boldsymbol{A}$ 是对称正定阵，即它的各阶顺序主子式 $D_k=\det(\boldsymbol{A}_k)>0, k=1,2,\cdots,n$，其中 $\boldsymbol{A}_k$ 为 $\boldsymbol{A}$ 的顺序主子矩阵. 从定理 6.3.3 可得如下定理.

定理 6.6.2 设 $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ 对称且其主子式 $D_k \neq 0$($k=1,2,\cdots,n$)，则存在惟一的单位下三角矩阵 $\boldsymbol{L}$ 和对角矩阵 $\boldsymbol{D}$，使

$$\boldsymbol{A} = \boldsymbol{LDL}^{\mathrm{T}}.$$

定理 6.6.3 设 $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ 为对称正定矩阵，则存在惟一的对角元为正的下三角矩阵 $\boldsymbol{L}$，使

$$\boldsymbol{A} = \boldsymbol{LL}^{\mathrm{T}}.$$

1. 楚列斯基分解法

我们称对称正定矩阵 $\boldsymbol{A}$ 的 $\boldsymbol{LL}^{\mathrm{T}}$ 分解为楚列斯基分解. 直接利用 $\boldsymbol{A}=\boldsymbol{LL}^{\mathrm{T}}$,即

$$\boldsymbol{A}=\begin{pmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{pmatrix}\begin{pmatrix} l_{11} & l_{21} & \cdots & l_{n1} \\ & l_{22} & \cdots & l_{n2} \\ & & \ddots & \vdots \\ & & & l_{nn} \end{pmatrix}$$

的矩阵乘法规则,且规定 $l_{ii}>0(i=1,2,\cdots,n)$,则可按列求得矩阵 $\boldsymbol{L}$ 元素的计算公式. 对于 $j=1,2,\cdots,n$,

$$l_{jj}=\left(a_{jj}-\sum_{k=1}^{j-1}l_{jk}^{2}\right)^{\frac{1}{2}};$$

$$l_{ij}=\left(a_{ij}-\sum_{k=1}^{j-1}l_{ik}l_{jk}\right)\Big/l_{jj},\quad i=j+1,\cdots,n.$$

求解方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 等价于顺序求解下列两个三角形方程组

$$\begin{cases}\boldsymbol{Ly}=\boldsymbol{b},\\ \boldsymbol{L}^{\mathrm{T}}\boldsymbol{x}=\boldsymbol{y},\end{cases}$$

即

$$y_i=\left(b_i-\sum_{k=1}^{i-1}l_{ik}y_k\right)\Big/l_{ii},\quad i=1,2,\cdots,n,$$

$$x_i=\left(b_i-\sum_{k=i+1}^{n}l_{ki}x_k\right)\Big/l_{ii},\quad i=n,n-1,\cdots,1.$$

利用 $\boldsymbol{A}$ 的楚列斯基分解式求解 $\boldsymbol{Ax}=\boldsymbol{b}$ 的方法称为**楚列斯基方法**或**平方根法**(square root method),其计算量约需 $\frac{n^3}{6}$ 次乘除法,还需进行 n 次开方运算. 平方根法的优点之一是不必选主元.

2. 改进的平方根法

为了避免开方运算,对于对称正定矩阵 $\boldsymbol{A}$ 采用定理 6.6.2 中的分解式 $\boldsymbol{A}=\boldsymbol{LDL}^{\mathrm{T}}$,即

$$A=\begin{pmatrix}1 & & & & & \\ l_{21} & 1 & & & & \\ l_{31} & l_{32} & 1 & & & \\ \vdots & \vdots & \vdots & \ddots & 1 & \\ l_{n1} & l_{n2} & l_{n3} & \cdots & l_{n,n-1} & 1\end{pmatrix}\begin{pmatrix}d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n\end{pmatrix}\begin{pmatrix}1 & l_{21} & l_{31} & \cdots & l_{n1} \\ & 1 & l_{32} & \cdots & l_{n2} \\ & & \ddots & & \vdots \\ & & & 1 & l_{n,n-1} \\ & & & & 1\end{pmatrix}$$

令 $\boldsymbol{T}=\boldsymbol{LD}$,我们有 $t_{ij}=l_{ij}d_j$,直接利用 $\boldsymbol{A}=\boldsymbol{TL}^{\mathrm{T}}$ 的矩阵乘法规则,有如下的按行计算 $\boldsymbol{L}$ 和 $\boldsymbol{T}$ 元素的公式:

$$d_1=a_{11}.$$

对于 $i=2,3,\cdots,n$,

(1) $t_{ij}=a_{ij}-\sum_{k=1}^{j-1}t_{ik}l_{jk}$, $j=1,2,\cdots,i-1$.

(2) $l_{ij}=t_{ij}/d_j$, $j=1,2,\cdots,i-1$.

(3) $d_i=a_{ii}-\sum_{k=1}^{i-1}t_{ik}l_{ik}$.

求解 $\boldsymbol{Ax}=\boldsymbol{b}$ 等价于顺序求解下列两个三角形方程组

$$\begin{cases}\boldsymbol{Ly}=\boldsymbol{b},\\ \boldsymbol{DL}^{\mathrm{T}}\boldsymbol{x}=\boldsymbol{y},\end{cases}$$

即

$$\begin{cases}y_1=b_1,\\ y_i=b_i-\sum_{k=1}^{i-1}l_{ik}y_k, \quad i=2,\cdots,n.\end{cases}$$

$$\begin{cases}x_n=y_n/d_n,\\ x_i=y_i/d_i-\sum_{k=i+1}^{n}l_{ki}x_k, \quad i=n-1,\cdots,1.\end{cases}$$

利用 $\boldsymbol{A}$ 的 $\boldsymbol{LDL}^{\mathrm{T}}$ 分解式求解 $\boldsymbol{Ax}=\boldsymbol{b}$ 的方法称为改进的平方根法,它与平方根法计算量差不多,但避免了开方运算.

6.7 带状方程组的解法

设 $n\times n$ 矩阵 $\boldsymbol{A}$ 满足对于 $j>i+q$ 时，有 $a_{ij}=0$，则称 $\boldsymbol{A}=(a_{ij})$ 具有上带宽 q，对于 $i>j+p$ 时有 $a_{ij}=0$，则称 $\boldsymbol{A}$ 具有下带宽 p，$\boldsymbol{A}$ 的带宽为 $p+q+1$. 当 $p=q$ 时，称 $\boldsymbol{A}$ 为等带宽.

定理 6.7.1 假定 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 存在 LU 分解 $\boldsymbol{A}=\boldsymbol{LU}$，如果 $\boldsymbol{A}$ 的上带宽为 q，下带宽为 p，则 $\boldsymbol{U}$ 的上带宽为 q，$\boldsymbol{L}$ 的下带宽为 p.

该定理说明：$\boldsymbol{L}(\boldsymbol{U})$ 具有与 $\boldsymbol{A}$ 相同的下(上)带宽.

推论 6.7.2 假定 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 为对称正定矩阵，且有带宽 $2m+1$，其楚列斯基分解 $\boldsymbol{A}=\boldsymbol{LL}^{\mathrm{T}}$，则下三角矩阵 $\boldsymbol{L}$ 的下带宽为 m.

6.7.1 三对角方程组的托马斯法和循环约简法

设 $\boldsymbol{Ax}=\boldsymbol{d}$ 是三对角方程组，即

$$\begin{pmatrix} b_1 & c_1 & & \\ a_2 & b_2 & \ddots & \\ & \ddots & \ddots & c_{n-1} \\ & & a_n & b_n \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix}=\begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_{n-1} \\ d_n \end{pmatrix}. \tag{6.7.1}$$

根据定理 6.3.3，如果 $\boldsymbol{A}$ 的顺序主子式皆非零，则存在惟一的 LU 分解. 因为 $\boldsymbol{A}$ 的带宽为 $2m+1=3$. 由定理 6.7.1，$\boldsymbol{L}(\boldsymbol{U})$ 具有与 $\boldsymbol{A}$ 相同的下(上)带宽 $m=1$. 所以 $\boldsymbol{A}$ 有如下形式的 LU 分解.

$$\boldsymbol{A}=\boldsymbol{LU}=\begin{pmatrix} 1 & & & & \\ l_2 & 1 & & & \\ & l_3 & \ddots & & \\ & & \ddots & 1 & \\ & & & l_n & 1 \end{pmatrix}\begin{pmatrix} u_1 & c_1 & & & \\ & u_2 & c_2 & & \\ & & \ddots & \ddots & \\ & & & u_{n-1} & c_{n-1} \\ & & & & u_n \end{pmatrix}.$$

根据矩阵的乘法规则，易求得 $\boldsymbol{L}$ 和 $\boldsymbol{U}$ 的元素，再求解 $\boldsymbol{Ly}=\boldsymbol{d}$

和 $\boldsymbol{Ux}=\boldsymbol{y}$,得到 $\boldsymbol{Ax}=\boldsymbol{b}$ 的解,称此求解三对角方程组的方法为**托马斯(Thomas)算法或追赶法**.

算法 6.7.3 托马斯算法

用三个一维数组 $\boldsymbol{a}=(0,a_2,a_3,\cdots,a_n)$,$\boldsymbol{b}=(b_1,b_2,\cdots,b_n)$,$\boldsymbol{c}=(c_1,c_2,\cdots,c_{n-1})$存放矩阵 $\boldsymbol{A}$ 的三条对角线元素.数组 $\boldsymbol{d}=(d_1,d_2,\cdots,d_n)$存放右端项.

(1) 求 $\boldsymbol{L}$ 和 $\boldsymbol{U}$ 的元素

① $u_1=b_1$

② 对于 $i=2,3,\cdots,n$

$l_i=a_i/u_{i-1}$

$u_i=b_i-l_ic_{i-1}$

(2) 求解 $\boldsymbol{Ly}=\boldsymbol{d}$ 和 $\boldsymbol{Ux}=\boldsymbol{y}$

① $y_1=d_1$

② $y_i=d_i-l_iy_{i-1}$, $i=2,3,\cdots,n$

③ $x_n=y_n/u_n$

④ $x_i=(y_i-c_ix_{i+1})/u_i$, $i=n-1,n-2,\cdots,1$

计算过程中,将 $u_1,\cdots,u_n$ 和 $l_2,\cdots,l_n$ 分别存放在数组 $\boldsymbol{b}$ 和 $\boldsymbol{a}$ 的相应位置,数组 $\boldsymbol{c}$ 不变.三角形方程组的解 $\boldsymbol{y}$ 和 $\boldsymbol{x}$ 先后存放在数组 $\boldsymbol{d}$ 的相应位置.最后 $\boldsymbol{Ax}=\boldsymbol{d}$ 的解 $\boldsymbol{x}$ 放在 $\boldsymbol{d}$ 中.

定理 6.7.4 设三对角方程组 $\boldsymbol{Ax}=\boldsymbol{d}$ 的系数矩阵满足:

(1) $|b_1|>|c_1|>0$;

(2) $|b_n|>|a_n|>0$;

(3) $|b_i|\geqslant|a_i|+|c_i|$,$a_ic_i\neq0$,$i=2,3,\cdots,n-1$.

则 $\boldsymbol{A}$ 非奇异且有

(i) $u_i\neq0$, $i=1,2,\cdots,n$;

(ii) $0<\dfrac{|c_i|}{|u_i|}<1$, $i=1,2,\cdots,n-1$;

(iii) $|b_i|-|a_i|<|u_i|<|b_i|+|a_i|$, $i=2,3,\cdots,n-1$.

因此，托马斯算法能顺利进行计算.

三对角方程组的**循环约简法**(cycle reduced method)是适合并行计算的方法. 为了方便，设方程组(6.7.1)中方程的数目 $n=2^l-1$，l 为一个正整数，方程组(6.7.1)相邻的三个方程写成

$$a_{i-1}x_{i-2}+b_{i-1}x_{i-1}+c_{i-1}x_i=d_{i-1}; \tag{6.7.2}$$

$$a_i x_{i-1}+b_i x_i+c_i x_{i+1}=d_i; \tag{6.7.3}$$

$$a_{i+1}x_i+b_{i+1}x_{i+1}+c_i x_{i+2}=d_{i+1}. \tag{6.7.4}$$

将方程(6.7.2)乘以 $-a_i/b_{i-1}$ 和方程(6.7.4)乘以 $-c_i/b_{i+1}$，并加到方程(6.7.3)，得到

$$\hat{a}_i x_{i-2}+\hat{b}_i x_i+\hat{c}_i x_{i+2}=\hat{d}_i,$$

其中

$$\hat{a}_i=-a_{i-1}\frac{a_i}{b_{i-1}},$$

$$\hat{b}_i=b_i-c_{i-1}\frac{a_i}{b_{i-1}}-a_{i+1}\frac{c_i}{b_{i+1}},$$

$$\hat{c}_i=-c_{i+1}\frac{c_i}{b_{i+1}},$$

$$\hat{d}_i=-\frac{a_i}{b_{i-1}}d_{i-1}+d_i-\frac{c_i}{b_{i+1}}d_{i+1}.$$

因此，如果 i 为偶数，我们得到关于未知量 $x_2,x_4,\cdots,x_{n-1}$ 的方程组

$$\begin{pmatrix} \hat{b}_2 & \hat{c}_2 & & & \\ \hat{a}_4 & \hat{b}_4 & \hat{c}_4 & & \\ & \ddots & \ddots & \ddots & \\ & & \hat{a}_{n-3} & \hat{b}_{n-3} & \hat{c}_{n-3} \\ & & & \hat{a}_{n-1} & \hat{b}_{n-1} \end{pmatrix} \begin{pmatrix} x_2 \\ x_4 \\ \vdots \\ x_{n-3} \\ x_{n-1} \end{pmatrix} = \begin{pmatrix} \hat{d}_2 \\ \hat{d}_4 \\ \vdots \\ \hat{d}_{n-3} \\ \hat{d}_{n-1} \end{pmatrix}. \tag{6.7.5}$$

而关于 $x_1,x_3,\cdots,x_n$ 的方程组为

$$\begin{pmatrix} b_1 & & & & \\ & b_3 & & & \\ & & \ddots & & \\ & & & b_{n-2} & \\ & & & & b_n \end{pmatrix} \begin{pmatrix} x_1 \\ x_3 \\ \vdots \\ x_{n-2} \\ x_n \end{pmatrix} = \begin{pmatrix} d_1 \\ d_3 \\ \vdots \\ d_{n-2} \\ d_n \end{pmatrix} - \begin{pmatrix} c_1 & & & \\ a_3 & c_3 & & \\ & \ddots & \ddots & \\ & & a_{n-2} & c_{n-2} \\ & & & a_n \end{pmatrix} \begin{pmatrix} x_2 \\ x_4 \\ \vdots \\ x_{n-3} \\ x_{n-1} \end{pmatrix}, \tag{6.7.6}$$

所以，若从方程组(6.7.5)解出 $x_2, x_4, \cdots, x_{n-1}$，则可从方程组(6.7.6)计算出 $x_1, x_3, \cdots, x_n$. 事实上方程组(6.7.5)也是三对角的，其阶数是 $2^{l-1}-1$. 可以用类似的方法继续约简它，直到约简到只剩 1 个方程. 求出未知数，再利用各步形成的方程组(6.7.6)，计算各个分量，称此方法为**循环约简法**.

如果 $n \neq 2^l - 1$，则约简过程的最后一步，得到的不是一个方程，而是含若干个方程的方程组，解出其分量后，再向后回代.

6.7.2 块三对角方程组的解法

设 $\boldsymbol{Ax}=\boldsymbol{b}$，其中 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ 是带宽为 $2m+1$ 的带状矩阵. 将 $\boldsymbol{A}$ 划分子块为块三对角形式，$\boldsymbol{x}$ 和 $\boldsymbol{b}$ 也作相应的划分，则 $\boldsymbol{Ax}=\boldsymbol{b}$ 的分块形式为

$$\begin{pmatrix} \boldsymbol{D}_1 & \boldsymbol{F}_1 & & \\ \boldsymbol{E}_1 & \boldsymbol{D}_2 & \ddots & \\ & \ddots & \ddots & \boldsymbol{F}_{p-1} \\ & & \boldsymbol{E}_{p-1} & \boldsymbol{D}_p \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_{p-1} \\ \boldsymbol{x}_p \end{pmatrix} = \begin{pmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \vdots \\ \boldsymbol{b}_{p-1} \\ \boldsymbol{b}_p \end{pmatrix}, \tag{6.7.7}$$

其中每个子块均是 $q\times q$ 矩阵，$q=n/p$，$\boldsymbol{b}_i, \boldsymbol{x}_i \in \mathbb{R}^q$，$i=1,2,\cdots,p$.

1. 块托马斯法

将块矩阵 $\boldsymbol{A}$ 作块 LU 分解：

$$
\boldsymbol{A}=\begin{pmatrix} \boldsymbol{I} & & & \\ \boldsymbol{L}_1 & \boldsymbol{I} & & \\ & \ddots & \ddots & \\ & & \boldsymbol{L}_{p-1} & \boldsymbol{I} \end{pmatrix}\begin{pmatrix} \boldsymbol{U}_1 & \boldsymbol{F}_1 & & & \\ & \boldsymbol{U}_2 & \boldsymbol{F}_2 & & \\ & & \ddots & \ddots & \\ & & & \boldsymbol{U}_{p-1} & \boldsymbol{F}_{p-1} \\ & & & & \boldsymbol{U}_p \end{pmatrix}=\boldsymbol{LU}. \tag{6.7.8}
$$

根据分块矩阵乘法规则,得

$$
\begin{cases} \boldsymbol{U}_1=\boldsymbol{D}_1; \\ \boldsymbol{L}_{i-1}\boldsymbol{U}_{i-1}=\boldsymbol{E}_{i-1}, \qquad \text{解出 } \boldsymbol{L}_{i-1}, \quad i=2,\cdots,p_j; \\ \boldsymbol{U}_i=\boldsymbol{D}_i-\boldsymbol{L}_{i-1}\boldsymbol{F}_{i-1}, \quad i=2,3,\cdots,p. \end{cases} \tag{6.7.9}
$$

当矩阵 $\boldsymbol{A}$ 的块三对角子矩阵 $\boldsymbol{A}_k$ 为

$$
\boldsymbol{A}_k=\begin{pmatrix} \boldsymbol{D}_1 & \boldsymbol{F}_1 & & & \\ \boldsymbol{E}_1 & \boldsymbol{D}_2 & \boldsymbol{F}_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \boldsymbol{E}_{k-2} & \boldsymbol{D}_{k-1} & \boldsymbol{F}_{k-1} \\ & & & \boldsymbol{E}_{k-1} & \boldsymbol{D}_k \end{pmatrix}, \quad k=1,2,\cdots,p,
$$

则当 $\boldsymbol{A}_k, k=1,2,\cdots,n$ 均为非奇异阵时,$\boldsymbol{U}_i$ 是非奇异的,块分解式(6.7.9)就能继续.

再求解方程组 $\boldsymbol{Ly}=\boldsymbol{b}$ 和 $\boldsymbol{Ux}=\boldsymbol{y}$,即

$$
\begin{cases} \boldsymbol{y}_1=\boldsymbol{b}_1; \\ \boldsymbol{y}_i=\boldsymbol{b}_i-\boldsymbol{L}_{i-1}\boldsymbol{y}_{i-1}, \quad i=2,\cdots,p. \end{cases}
$$

$$
\begin{cases} \boldsymbol{U}_p\boldsymbol{x}_p=\boldsymbol{y}_p, \qquad \text{解出 } \boldsymbol{x}_p; \\ \boldsymbol{U}_i\boldsymbol{x}_i=\boldsymbol{y}_i-\boldsymbol{F}_i\boldsymbol{x}_{i+1}, \quad \text{解出 } \boldsymbol{x}_i, \quad i=p-1,\cdots,1. \end{cases}
$$

称以上方法为**块三对角方程组的托马斯法或块追赶法**.

2. 分块循环约简法

假定 $\boldsymbol{Ax}=\boldsymbol{b}$ 中的 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 具有如下形式:

$$
\boldsymbol{A}=\begin{pmatrix}\boldsymbol{D} & \boldsymbol{F} & & & \\ \boldsymbol{F} & \boldsymbol{D} & \boldsymbol{F} & & \\ & \ddots & \ddots & \ddots & \\ & & \boldsymbol{F} & \boldsymbol{D} & \boldsymbol{F} \\ & & & \boldsymbol{F} & \boldsymbol{D}\end{pmatrix}, \tag{6.7.10}
$$

其中 $\boldsymbol{F}$ 和 $\boldsymbol{D}$ 均是 $q\times q$ 矩阵 $\left(q=\dfrac{n}{p}\right)$，且满足 $\boldsymbol{DF}=\boldsymbol{FD}$，同时假定 $p=2^k-1$. 这些条件在许多重要应用中是成立的.

循环约简法的基本思想是将问题的维数反复地减半. 块三对角方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 的相邻的三个方程写成

$$
\begin{aligned}
\boldsymbol{F}\boldsymbol{x}_{i-2}+\boldsymbol{D}\boldsymbol{x}_{i-1}+\boldsymbol{F}\boldsymbol{x}_i&=\boldsymbol{b}_{i-1};\\
\boldsymbol{F}\boldsymbol{x}_{i-1}+\boldsymbol{D}\boldsymbol{x}_{i}+\boldsymbol{F}\boldsymbol{x}_{i+1}&=\boldsymbol{b}_{i};\\
\boldsymbol{F}\boldsymbol{x}_{i}+\boldsymbol{D}\boldsymbol{x}_{i+1}+\boldsymbol{F}\boldsymbol{x}_{i+2}&=\boldsymbol{b}_{i+1}.
\end{aligned}
$$

分别用 $\boldsymbol{F},-\boldsymbol{D},\boldsymbol{F}$ 乘每个方程，三个方程相加后，得

$$
\boldsymbol{F}^{(1)}\boldsymbol{x}_{i-2}+\boldsymbol{D}^{(1)}\boldsymbol{x}_i+\boldsymbol{F}^{(1)}\boldsymbol{x}_{i+2}=\boldsymbol{b}_i^{(1)},
$$

其中 $\boldsymbol{F}^{(1)}=\boldsymbol{F}^2,\boldsymbol{D}^{(1)}=2\boldsymbol{F}^2-\boldsymbol{D}^2,\boldsymbol{b}_i^{(1)}=\boldsymbol{F}(\boldsymbol{b}_{i-1}+\boldsymbol{b}_{i+1})-\boldsymbol{D}\boldsymbol{b}_i,\boldsymbol{x}_i\in\mathbb{R}^q,i=1,2,\cdots,p$. 若 i 为偶数，可得到一个约简后的关于 $\boldsymbol{x}_2,\boldsymbol{x}_4,\cdots,\boldsymbol{x}_{p-1}$ 的块三对角方程组

$$
\begin{pmatrix}\boldsymbol{D}^{(1)} & \boldsymbol{F}^{(1)} & & \\ \boldsymbol{F}^{(1)} & \boldsymbol{D}^{(1)} & \ddots & \\ & \ddots & \ddots & \boldsymbol{F}^{(1)} \\ & & \boldsymbol{F}^{(1)} & \boldsymbol{D}^{(1)}\end{pmatrix}\begin{pmatrix}\boldsymbol{x}_2\\ \boldsymbol{x}_4\\ \vdots\\ \boldsymbol{x}_{p-1}\end{pmatrix}=\begin{pmatrix}\boldsymbol{b}_2^{(1)}\\ \boldsymbol{b}_4^{(1)}\\ \vdots\\ \boldsymbol{b}_{p-1}^{(1)}\end{pmatrix}, \tag{6.7.11}
$$

其中 $\boldsymbol{D}^{(1)}\boldsymbol{F}^{(1)}=\boldsymbol{F}^{(1)}\boldsymbol{D}^{(1)}$，以类似的方法继续约简方程组(6.7.11)，直到只含一个块方程. 求解这个 $q\times q$ 方程组，再利用约简各步形成的方程组(6.7.11)，计算各个编号为偶数的解向量 $\boldsymbol{x}$ 的分向量 $\boldsymbol{x}_2,\boldsymbol{x}_4,\cdots,\boldsymbol{x}_{p-1}$. 然后再用原方程组中编号为奇数的方程，分别计算 $\boldsymbol{x}_1,\boldsymbol{x}_3,\cdots,\boldsymbol{x}_p$. 称这种方法为**块循环约简法**(block cycle reduced method).

算法 6.7.5　块循环约简法

给定方程组 $\boldsymbol{Ax}=\boldsymbol{b}$，其中 $\boldsymbol{A}$ 有形如式(6.7.10)的块三对角矩阵. $\boldsymbol{D},\boldsymbol{F}\in\mathbb{R}^{q\times q}$，$\boldsymbol{A}\in\mathbb{R}^{n\times n}$，$n=pq$，$p=2^k-1$. 令 $\boldsymbol{D}^{(0)}=\boldsymbol{D}$，$\boldsymbol{F}^{(0)}=\boldsymbol{F}$，$\boldsymbol{b}^{(0)}=\boldsymbol{b}$. 用块循环约简法计算 $\boldsymbol{x}$.

(1) 循环约简过程

　　对于 $l=1,2,\cdots,k-1$

　　　　$\boldsymbol{F}^{(l)}=(\boldsymbol{F}^{(l-1)})^2$

　　　　$\boldsymbol{D}^{(l)}=2\boldsymbol{F}^{(l)}-(\boldsymbol{D}^{(l-1)})^2$

　　　　$r=2^l$

　　　　对于 $j=1,2,\cdots,2^{k-l}-1$

　　　　　　$\boldsymbol{b}_{jr}^{(l)}=\boldsymbol{F}^{(l-1)}(\boldsymbol{b}_{jr-\frac{r}{2}}^{(l-1)}+\boldsymbol{b}_{jr+\frac{r}{2}}^{(l-1)})-\boldsymbol{D}^{(l-1)}\boldsymbol{b}_{jr}^{(l-1)}$

　　　　　　j 尾

　　　　l 尾

(2) 计算 $\boldsymbol{x}_i$

　　① 从 $\boldsymbol{D}^{(k-1)}\boldsymbol{x}_{2^{k-1}}=\boldsymbol{b}_{2^{k-1}}^{(k-1)}$ 解出 $\boldsymbol{x}_{2^{k-1}}$

　　② 对于 $l=k-2,k-3,\cdots,0$

　　　　$r=2^l$.

　　　　对于 $j=1,2,\cdots,2^{k-l-1}$

　　　　(i) 如果 $j=1$，则 $\boldsymbol{c}=\boldsymbol{b}_{(2j-1)r}^{(l)}-\boldsymbol{F}^{(l)}\boldsymbol{x}_{2jr}$

　　　　　　否则如果 $j=2^{k-l-1}$，则

　　　　　　　$\boldsymbol{c}=\boldsymbol{b}_{(2j-1)r}^{(l)}-\boldsymbol{F}^{(l)}\boldsymbol{x}_{(2j-2)r}$

　　　　　　否则 $\boldsymbol{c}=\boldsymbol{b}_{(2j-1)r}^{(l)}-\boldsymbol{F}^{(l)}(\boldsymbol{x}_{2jr}+\boldsymbol{x}_{(2j-2)r})$

　　　　(ii) 从 $\boldsymbol{D}^{(l)}\boldsymbol{x}_{(2j-1)r}=\boldsymbol{c}$，解出 $\boldsymbol{x}_{(2j-1)r}$

　　　　　　j 尾

　　　　l 尾

该算法的工作量的大小很大程度依赖于 $q\times q$ 矩阵 $\boldsymbol{D}^{(l)}$ 和 $\boldsymbol{F}^{(l)}$ 的稀疏性. 在较坏的情况下，即当它们是满阵时，总的浮点运算量为 $\log(p)q^3$. 算法的存储量不大.

3. 约翰逊算法

约翰逊(Johnsson)算法是块三对角方程组(6.7.7)的一种约简法.通常方程组(6.7.7)中的 $\boldsymbol{E}_i$ 为上三角矩阵,$\boldsymbol{F}_i$ 为下三角矩阵.

将对角块 $\boldsymbol{D}_i$ 划分为

$$\boldsymbol{D}_i=\begin{pmatrix}\boldsymbol{D}_{i1} & \boldsymbol{D}_{i2}\\ \boldsymbol{D}_{i3} & \boldsymbol{D}_{i4}\end{pmatrix},\quad i=1,2,\cdots,p,$$

其中 $\boldsymbol{D}_{i4}$ 是 $s\times s$ 矩阵,$\boldsymbol{D}_{i1}$ 是 $(q-s)\times(q-s)$ 矩阵,其他子块 $\boldsymbol{E}_i,\boldsymbol{F}_i$ 和向量 $\boldsymbol{x}_i,\boldsymbol{b}_i$ 也相应地划分为

$$\boldsymbol{E}_i=\begin{pmatrix}\boldsymbol{0} & \boldsymbol{E}_{i1}\\ \boldsymbol{0} & \boldsymbol{E}_{i2}\end{pmatrix},\quad \boldsymbol{F}_i=\begin{pmatrix}\boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{F}_{i1} & \boldsymbol{F}_{i2}\end{pmatrix},\quad \boldsymbol{x}_i=\begin{pmatrix}\boldsymbol{x}_{i1}\\ \boldsymbol{x}_{i2}\end{pmatrix},\quad \boldsymbol{b}_i=\begin{pmatrix}\boldsymbol{b}_{i1}\\ \boldsymbol{b}_{i2}\end{pmatrix}.$$

如果 $2s<q$,则 $\boldsymbol{F}_{i2}$ 和 $\boldsymbol{E}_{i2}$ 为零矩阵.

将块三对角方程组(6.7.7)两边乘块对角矩阵

$$\mathrm{diag}(\boldsymbol{D}_{11}^{-1},\boldsymbol{I},\boldsymbol{D}_{21}^{-1},\boldsymbol{I},\cdots,\boldsymbol{D}_{p1}^{-1},\boldsymbol{I})$$

得到新的方程组(以 $p=3$ 为例)

$$\begin{pmatrix}\boldsymbol{I} & \boldsymbol{D}_{12}^1 & & & & \\ \boldsymbol{D}_{13} & \boldsymbol{D}_{14} & \boldsymbol{F}_{11} & \boldsymbol{F}_{12} & & \\ & \boldsymbol{E}_{21}^1 & \boldsymbol{I} & \boldsymbol{D}_{22}^1 & & \\ & \boldsymbol{E}_{22} & \boldsymbol{D}_{23} & \boldsymbol{D}_{24} & \boldsymbol{F}_{21} & \boldsymbol{F}_{22}\\ & & & \boldsymbol{E}_{31}^1 & \boldsymbol{I} & \boldsymbol{D}_{32}^1\\ & & & \boldsymbol{E}_{32} & \boldsymbol{D}_{33} & \boldsymbol{D}_{34}\end{pmatrix}\begin{pmatrix}\boldsymbol{x}_{11}\\ \boldsymbol{x}_{12}\\ \boldsymbol{x}_{21}\\ \boldsymbol{x}_{22}\\ \boldsymbol{x}_{31}\\ \boldsymbol{x}_{32}\end{pmatrix}=\begin{pmatrix}\boldsymbol{b}_{11}^1\\ \boldsymbol{b}_{12}\\ \boldsymbol{b}_{21}^1\\ \boldsymbol{b}_{22}\\ \boldsymbol{b}_{31}^1\\ \boldsymbol{b}_{32}\end{pmatrix},\tag{6.7.12}$$

其中 $\boldsymbol{D}_{i1}\boldsymbol{D}_{i2}^1=\boldsymbol{D}_{i2},\boldsymbol{D}_{i1}\boldsymbol{E}_{i1}^1=\boldsymbol{E}_{i1},\boldsymbol{D}_{i1}\boldsymbol{b}_{i1}^1=\boldsymbol{b}_{i1}$.在方程组(6.7.12)中,用 $-\boldsymbol{D}_{13}$ 左乘第 1 个块方程,$-\boldsymbol{F}_{11}$ 左乘第 3 个块方程,都加到第 2 个块方程中,又得到一个新的块方程

$$\boldsymbol{D}_{14}^1\boldsymbol{x}_{12}+\boldsymbol{F}_{12}^1\boldsymbol{x}_{22}=\boldsymbol{b}_{12}^1,$$

其中 $\boldsymbol{D}_{14}^1=\boldsymbol{D}_{14}-\boldsymbol{D}_{13}\boldsymbol{D}_{12}^1-\boldsymbol{F}_{11}\boldsymbol{E}_{21}^1,\boldsymbol{F}_{12}^1=\boldsymbol{F}_{12}-\boldsymbol{F}_{11}\boldsymbol{D}_{22}^1,\boldsymbol{b}_{12}^1=\boldsymbol{b}_{12}-\boldsymbol{D}_{13}\boldsymbol{b}_{11}^1-\boldsymbol{F}_{11}\boldsymbol{b}_{21}^1$.再用 $-\boldsymbol{D}_{23}$ 左乘第 3 个块方程,$-\boldsymbol{F}_{21}$ 左乘第 5 个块方程,都加到第 4 个块方程中,最后用 $-\boldsymbol{D}_{33}$ 左乘第 5 个块方程加

到第 6 个块方程中，得到如下的块方程组：

$$\begin{pmatrix} \boldsymbol{I} & \boldsymbol{D}_{12}^1 & & & & \\ & \boldsymbol{D}_{14}^1 & \boldsymbol{0} & \boldsymbol{F}_{12}^1 & & \\ & \boldsymbol{E}_{21}^1 & \boldsymbol{I} & \boldsymbol{D}_{22}^1 & & \\ & \boldsymbol{E}_{22}^1 & \boldsymbol{0} & \boldsymbol{D}_{24}^1 & \boldsymbol{0} & \boldsymbol{F}_{22}^1 \\ & & & \boldsymbol{E}_{31}^1 & \boldsymbol{I} & \boldsymbol{D}_{32}^1 \\ & & & \boldsymbol{E}_{32}^1 & \boldsymbol{0} & \boldsymbol{D}_{34}^1 \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_{11} \\ \boldsymbol{x}_{12} \\ \boldsymbol{x}_{21} \\ \boldsymbol{x}_{22} \\ \boldsymbol{x}_{31} \\ \boldsymbol{x}_{32} \end{pmatrix} = \begin{pmatrix} \boldsymbol{b}_{11}^1 \\ \boldsymbol{b}_{12}^1 \\ \boldsymbol{b}_{21}^1 \\ \boldsymbol{b}_{22}^1 \\ \boldsymbol{b}_{31}^1 \\ \boldsymbol{b}_{32}^1 \end{pmatrix}, \quad (6.7.13)$$

其中的第 2,4,6 个块方程为约简方程

$$\begin{pmatrix} \boldsymbol{D}_{14}^1 & \boldsymbol{F}_{12}^1 & \\ \boldsymbol{E}_{22}^1 & \boldsymbol{D}_{24}^1 & \boldsymbol{F}_{22}^1 \\ & \boldsymbol{E}_{32}^1 & \boldsymbol{D}_{34}^1 \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_{12} \\ \boldsymbol{x}_{22} \\ \boldsymbol{x}_{32} \end{pmatrix} = \begin{pmatrix} \boldsymbol{b}_{12}^1 \\ \boldsymbol{b}_{22}^1 \\ \boldsymbol{b}_{32}^1 \end{pmatrix}. \quad (6.7.14)$$

其中系数矩阵中的各块矩阵和右端项均可从约简过程中建立它们的计算公式. 从式(6.7.14)解出 $\boldsymbol{x}_{i2}(i=1,2,3)$便可从式(6.7.13)的奇数序号方程求出 $\boldsymbol{x}_{i1}$，即

$$\boldsymbol{x}_{i1} = \boldsymbol{b}_{i1}^1 - \boldsymbol{D}_{i2}^1 \boldsymbol{x}_{i2} - \boldsymbol{E}_{i1}^1 \boldsymbol{x}_{i-1,2}, \quad i = 1,2,3, \quad (6.7.15)$$

其中 $i=1$ 时，$\boldsymbol{E}_{11}^1=\boldsymbol{0}$.

以上过程称为**约翰逊约简方法**. 可推广到 p 个块三对角方程组(6.7.7).

约翰逊算法的步骤如下：

(1) 求解方程组 $\boldsymbol{D}_{i1}\boldsymbol{D}_{i2}^1=\boldsymbol{D}_{i2}$，$\boldsymbol{D}_{i1}\boldsymbol{E}_{i1}^1=\boldsymbol{E}_{i1}$，$\boldsymbol{D}_{i1}\boldsymbol{b}_{i1}^1=\boldsymbol{b}_{i1}$只需对 $\boldsymbol{D}_{i1}$作 1 次 LU 分解，分别求得 $\boldsymbol{D}_{i2}^1$，$\boldsymbol{E}_{i1}^1$ 和 $\boldsymbol{b}_{i1}^1$，$i=1,2,\cdots,p$(其中 $\boldsymbol{E}_{11}^1=\boldsymbol{0}$)

(2) 计算 $\boldsymbol{D}_{i4}^1$，$\boldsymbol{E}_{i2}$，$\boldsymbol{F}_{i2}^1$和 $\boldsymbol{b}_{i2}^1$，$i=1,2,\cdots,p$(其中 $\boldsymbol{F}_{p2}^1=\boldsymbol{0}$)

(3) 求解块方程组(如式(6.7.14))，得 $\boldsymbol{x}_{i2}$，$i=1,2,\cdots,p$

(4) 从式(6.7.15)计算 $\boldsymbol{x}_{i1}$，$i=1,2,\cdots,p$

其中 $i=1$ 时，$\boldsymbol{E}_{11}^1=\boldsymbol{0}$.

约翰逊算法的第 1,2,4 步很容易实行并行计算.

6.7.3 带状方程组的列主元消去法

考虑求解带宽为 $2m+1$ 的一般型方程组

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},$$

6.4 节已给出一般列主元消去法的描述，它也适用于带状方程组. 这里利用 $\boldsymbol{A}$ 的带状结构，改进算法，达到节省计算机时间和存储单元的目的.

以五阶三对角方程组(即 $2m+1=5, m=1$)为例，陈述带状矩阵的存储方式以及相应的列主元消元过程. 设 $\boldsymbol{A}$ 和 $\boldsymbol{b}$ 为

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} & & & \\ a_{21} & a_{22} & a_{23} & & \\ & a_{32} & a_{33} & a_{34} & \\ & & a_{43} & a_{44} & a_{45} \\ & & & a_{54} & a_{55} \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{pmatrix}.$$

将 $\boldsymbol{A}$ 带区内的元素按行存放在一个 n 行 $2m+1$ 列的矩形数组 $\boldsymbol{C}(1:n, 1:2m+1)$ 中. 其中前 m 行和后 m 行每行带区内元素不足 $2m+1$ 的，在行的后面增添零元素补齐. 有如下的增广数组

$$(\boldsymbol{C} \mid \boldsymbol{b}) = \left(\begin{array}{ccc|c} a_{11} & a_{12} & 0 & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{32} & a_{33} & a_{34} & b_3 \\ a_{43} & a_{44} & a_{45} & b_4 \\ a_{54} & a_{55} & 0 & b_5 \end{array}\right).$$

每步消元涉及的是 $m+1$ 行元素.

第一步消元：在前 $m+1$ 行的第 1 列选主元，确定 i_1，使 $|a_{i_1,1}| = \max\limits_{1\leqslant i\leqslant m+1} |a_{i,1}|$. 将第 i_1，1 行交换. 记第 1 次消元的主元为 $\bar{a}_{11}$，将第 1 行的其余元素除以 $\bar{a}_{11}$，得 $\tilde{a}_{12}$，$\tilde{a}_{13}$ 和 $\tilde{b}_1$. 对第 2 行(一般情况第 2 行到第 $m+1$ 行)消元，使它们的第 1 列元素为 0. 再将 $\boldsymbol{C}$(不包括 $\boldsymbol{b}$)

的第 2 行(一般情况第 2 行到第 $m+1$ 行)的元素依次向左移一位,每行的最后一个元素补成 0,即

$$(\boldsymbol{C} \mid \boldsymbol{b})=\left(\begin{array}{ccc:c}\bar{a}_{11} & \tilde{a}_{12} & \tilde{a}_{13} & \tilde{b}_1 \\ \hat{a}_{22} & \hat{a}_{23} & 0 & \hat{b}_2 \\ a_{32} & a_{33} & a_{34} & b_3 \\ a_{43} & a_{44} & a_{45} & b_4 \\ a_{54} & a_{55} & 0 & b_5\end{array}\right).$$

同理进行第 2 次选主元和第 2 次消元;第 3 次选主元和第 3 次消元.结果依次为

$$(\boldsymbol{C} \mid \boldsymbol{b})=\left(\begin{array}{ccc:c}\bar{a}_{11} & \tilde{a}_{12} & \tilde{a}_{13} & \tilde{b}_1 \\ \bar{a}_{22} & \tilde{a}_{23} & \tilde{a}_{24} & \tilde{b}_2 \\ \hat{a}_{33} & \hat{a}_{34} & 0 & \hat{b}_3 \\ a_{43} & a_{44} & a_{45} & b_4 \\ a_{54} & a_{55} & 0 & b_5\end{array}\right), \quad (\boldsymbol{C} \mid \boldsymbol{b})=\left(\begin{array}{ccc:c}\bar{a}_{11} & \tilde{a}_{12} & \tilde{a}_{13} & \tilde{b}_1 \\ \bar{a}_{22} & \tilde{a}_{23} & \tilde{a}_{24} & \tilde{b}_2 \\ \bar{a}_{33} & \tilde{a}_{34} & \tilde{a}_{35} & \tilde{b}_3 \\ \hat{a}_{44} & \hat{a}_{45} & 0 & \hat{b}_4 \\ a_{54} & a_{55} & 0 & b_5\end{array}\right).$$

最后一次选主元和消元后,得到

$$(\boldsymbol{C} \mid \boldsymbol{b})=\left(\begin{array}{ccc:c}\bar{a}_{11} & \tilde{a}_{12} & \tilde{a}_{13} & \tilde{b}_1 \\ \bar{a}_{22} & \tilde{a}_{23} & \tilde{a}_{24} & \tilde{b}_2 \\ \bar{a}_{33} & \tilde{a}_{34} & \tilde{a}_{35} & \tilde{b}_3 \\ \bar{a}_{44} & \tilde{a}_{45} & 0 & \tilde{b}_4 \\ \hat{a}_{55} & 0 & 0 & \hat{b}_5\end{array}\right).$$

数组 $\boldsymbol{C}$ 的第 1 列是所有主元素.若出现零主元,则消元过程终止.若上述消元过程不中断,则得到一个与原方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 等价的问题:

$$\begin{pmatrix}1 & \tilde{a}_{12} & \tilde{a}_{13} & & \\ & 1 & \tilde{a}_{23} & \tilde{a}_{24} & \\ & & 1 & \tilde{a}_{34} & \tilde{a}_{35} \\ & & & 1 & \tilde{a}_{45} \\ & & & & \hat{a}_{55}\end{pmatrix}\begin{pmatrix}x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5\end{pmatrix}=\begin{pmatrix}\tilde{b}_1 \\ \tilde{b}_2 \\ \tilde{b}_3 \\ \tilde{b}_4 \\ \hat{b}_5\end{pmatrix},$$

回代得 $\boldsymbol{Ax}=\boldsymbol{b}$ 的解 $\boldsymbol{x}=(x_1,x_2,\cdots,x_5)^{\mathrm{T}}$.

具体算法如下,整个消元和回代过程在$(\boldsymbol{C}|\boldsymbol{b})$数组中进行.

算法 6.7.6　带状方程组列主元素消去法

求解 n 阶带宽为 $2m+1$ 的带状方程组 $\boldsymbol{Ax}=\boldsymbol{b}$. 矩阵 $\boldsymbol{A}$ 以数组 $\boldsymbol{C}(1:n,1:2m+1)$的形式存放,解 $\boldsymbol{x}$ 存放在 $\boldsymbol{b}$ 中.

(1) 消元过程

对于 $k=1,2,\cdots,n-1$

① 选主元,确定 i_k,使

$$|c_{i_k,1}|=\max_{k\leqslant i\leqslant k+m}|c_{i1}|$$

② 如果 $i_k=k$,则转(3)

否则$c_{kj}\leftrightarrow c_{i_k,j}(j=1,2,\cdots,2m+1)$

$b_k\leftrightarrow b_{i_k}$

③ 计算

$$c_{kj}:=c_{kj}/c_{k1},\quad j=2,3,\cdots,2m+1$$

$$b_k:=b_k/c_{k1}$$

$$r:=\min\{k+m,n\}$$

④ 计算

$$b_i:=b_i-c_{i1}b_k,\quad i=k+1,\cdots,r$$

⑤ 计算

$$c_{i,j-1}:=c_{ij}-c_{i1}c_{kj},\quad i=k+1,\cdots,r,\quad j=2,3,\cdots,2m+1$$

⑥ 冲零

$$c_{i,2m+1}:=0,\quad i=k+1,\cdots,r$$

(2) 回代过程

计算

$$b_n\leftarrow x_n=b_n/c_{n1}$$

$$b_i\leftarrow x_i=b_i-\sum_{j=2}^{2m+1}c_{ij}b_{i+j-1},\quad i=n-1,n-2,\cdots,1$$

例 6.7.7 用算法 6.7.6 求解带状方程组 $\boldsymbol{Ax}=\boldsymbol{b}$,即

$$\begin{pmatrix} 7.5 & 3.5 & & \\ 18 & 33 & 4.1 & \\ & 9 & 103 & -1.5 \\ & & 3.7 & 19.3 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}=\begin{pmatrix} 11 \\ 55.1 \\ 110.5 \\ 23 \end{pmatrix},$$

这里 $n=4,m=1$. 数组 $\boldsymbol{C}(1:4,1:3)$及 $\boldsymbol{b}$ 为

$$(\boldsymbol{C}\mid\boldsymbol{b})=\left(\begin{array}{ccc:c} 7.5 & 3.5 & 0 & 11 \\ 18 & 33 & 4.1 & 55.1 \\ 9 & 103 & -1.5 & 110.5 \\ 3.7 & 19.3 & 0 & 23 \end{array}\right).$$

第一次消元在前两行之间进行,主元素为 18,第 1,2 行交换,消元后结果为

$$(\boldsymbol{C}\mid\boldsymbol{b})=\left(\begin{array}{ccc:c} 18 & 1.8\dot{3} & 0.22\dot{7} & 3.06\dot{1} \\ -10.25 & -1.708\dot{3} & 0 & -11.958\dot{3} \\ 9 & 103 & -1.5 & 110.5 \\ 3.7 & 19.3 & 0 & 23 \end{array}\right).$$

第二次消元在第 2,3 行进行. 主元素为-10.25,消元结果为

$$(\boldsymbol{C}\mid\boldsymbol{b})=\left(\begin{array}{ccc:c} 18 & 1.8\dot{3} & 0.22\dot{7} & 3.06\dot{1} \\ -10.25 & 0.1\dot{6} & 0 & 1.1\dot{6} \\ 101.5 & -1.5 & 0 & 100 \\ 3.7 & 19.3 & 0 & 23 \end{array}\right).$$

第三次消元在第 3,4 行进行,主元素为 101.5,最后消元结果为

$$(\boldsymbol{C}\mid\boldsymbol{b})=\left(\begin{array}{ccc:c} 18 & 1.8\dot{3} & 0.22\dot{7} & 3.06\dot{1} \\ -10.25 & 0.1\dot{6} & 0 & 1.1\dot{6} \\ 101.5 & -0.014778325 & 0 & 0.985221674 \\ 19.3546798 & 0 & 0 & 19.3546798 \end{array}\right).$$

数组 $\boldsymbol{C}$ 的第 1 列存放着全部主元素. 最后得到一个与原方程组等价的上三角形方程组

$$\begin{pmatrix} 1 & 1.8\dot{3} & 0.22\dot{7} & 0 \\ & 1 & 0.1\dot{6} & 0 \\ & & 1 & -0.014778325 \\ & & & 19.3546798 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 3.06\dot{1} \\ 1.1\dot{6} \\ 0.985221674 \\ 19.3546798 \end{pmatrix},$$

回代后得方程组的解 $\boldsymbol{x}=(1,1,1,1)^{\mathrm{T}}$.

6.7.4 对称正定带状方程组的解法

设 $\boldsymbol{A}$ 是 n 阶对称正定带状矩阵，带宽为 $2m+1$. 要解方程组

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \tag{6.7.16}$$

则 $\boldsymbol{A}$ 不必选主元可分解为

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^{\mathrm{T}}, \tag{6.7.17}$$

其中 $\boldsymbol{L}$ 为下半带宽为 m 的带状阵

$$\boldsymbol{L} = \begin{pmatrix} l_{11} & & & & & \\ l_{21} & l_{22} & & & & \\ \vdots & \ddots & \ddots & & & \\ l_{m+1,1} & \cdots & l_{m+1,m} & l_{m+1,m+1} & & \\ & \ddots & & \ddots & \ddots & \\ & & l_{n,n-m} & \cdots & l_{n,n-1} & l_{nn} \end{pmatrix},$$

即当 $i<j$ 或 $i-j>m$ 时，$l_{ij}=0$. $\boldsymbol{D}$ 为对角阵，其对角线元素为 $d_{ii}=1/l_{ii}$, $i=1,2,\cdots,n$.

由式(6.7.17)，直接用矩阵乘法规则，推出计算 $\boldsymbol{L}$ 元素的如下公式：

$$l_{ij} = a_{ij} - \sum_{k=r}^{j-1} l_{ik} l_{jk} / l_{kk}, \quad j = r, \cdots, i, \quad i = 1,2,\cdots,n \tag{6.7.18}$$

其中

$$r = \begin{cases} 1, & i \leqslant m+1, \\ i-m, & i > m+1. \end{cases}$$

同时对式(6.7.16)的右端项 b 作如下分解：

$$\boldsymbol{b}=\boldsymbol{L}\boldsymbol{D}\tilde{\boldsymbol{b}}, \tag{6.7.19}$$

直接利用矩阵和向量乘法规则，得到 $\tilde{\boldsymbol{b}}$ 的元素的计算公式

$$\tilde{b}_i=b_i-\sum_{j=r}^{i-1}l_{ij}\tilde{b}_j/l_{jj},\quad i=1,2,\cdots,n. \tag{6.7.20}$$

将 $\boldsymbol{A}$ 和 $\boldsymbol{b}$ 的分解式(6.7.17)和式(6.7.19)代入式(6.7.16)，可表示成等价的方程组

$$\boldsymbol{L}^{\mathrm{T}}\boldsymbol{x}=\tilde{\boldsymbol{b}}, \tag{6.7.21}$$

由此得出式(6.7.16)解的计算公式

$$x_i=\left(\tilde{b}_i-\sum_{j=i+1}^{t}l_{ji}\tilde{b}_j\right)\Big/l_{ii},\quad i=n,n-1,\cdots,1, \tag{6.7.22}$$

其中

$$t=\begin{cases}n, & i>n-m-1,\\ i+m, & i\leqslant n-m-1.\end{cases}$$

因为 $\boldsymbol{A}$ 对称，只需存放 $\boldsymbol{A}$ 的下半带区(包括对角线元素)于一个矩形数组 $\boldsymbol{C}(1:n,1:m+1)$，以六阶对称五对角矩阵为例，形式如下

$$\boldsymbol{C}=\begin{pmatrix}0 & 0 & a_{11}\\ 0 & a_{21} & a_{22}\\ a_{31} & a_{32} & a_{33}\\ a_{42} & a_{43} & a_{44}\\ a_{53} & a_{54} & a_{55}\\ a_{64} & a_{65} & a_{66}\end{pmatrix},$$

其中 $\boldsymbol{C}$ 的每 1 列是 $\boldsymbol{A}$ 的每条对角线元素，且左上角添零. a_{ij} 在 $\boldsymbol{C}$ 的位置如下：

$$c_{i,j-i+m+1}=a_{ij},\quad \begin{array}{l}\text{当 } i\leqslant m+1 \text{ 时}, j=1,2,\cdots,i;\\ \text{当 } i>m+1 \text{ 时}, j=i-m,\cdots,i.\end{array}$$

根据计算公式(6.7.18)、式(6.7.20)和式(6.7.22)，可构成如

下算法.

算法 6.7.8 带宽为 $2m+1$ 的对称正定方程组的解法.

给定 n 阶带宽为 $2m+1$ 的对称正定带状方程组 $\boldsymbol{Ax}=\boldsymbol{b}$. 矩阵 $\boldsymbol{A}$ 的下半带区(包括对角线元素)以数组 $c(1:n,1:m+1)$ 的形式存放. $\boldsymbol{L}$ 存放在 $\boldsymbol{C}$ 的位置,解 $\boldsymbol{x}$ 存放在 $\boldsymbol{b}$ 的位置.

(1) 计算 $\boldsymbol{L}$ 和 $\tilde{\boldsymbol{b}}$ 的过程

对于 $i=1,2,\cdots,n$

① 计算

$$c_{i,j-i+m+1} \leftarrow l_{ij} = c_{i,j-i+m+1} - \sum_{k=r}^{j-1} c_{i,k-i+m+1} c_{j,k-j+m+1} / c_{k,m+1}$$

其中,当 $i \leqslant m+1$ 时 $r=1, j=1,2,\cdots,i$; 当 $i>m+1$ 时 $r=i-m, j=i-m,\cdots,i$.

② 计算

$$b_i \leftarrow \tilde{b}_i = b_i - \sum_{j=r}^{i-1} c_{i,j-i+m+1} b_j / c_{j,m+1}$$

其中,当 $i \leqslant m+1$ 时,$r=1$; 当 $i>m+1$ 时,$r=i-m$.

(2) 求解 $\boldsymbol{L}^{\mathrm{T}}\boldsymbol{x}=\tilde{\boldsymbol{b}}$ 的过程

对于 $i=n,n-1,\cdots,1$

$$b_i \leftarrow x_i = \left(b_i - \sum_{j=i+1}^{t} c_{j,i-j+m+1} b_j\right) \Big/ c_{i,m+1}$$

其中,当 $i>n-m-1$ 时,$t=n$; 当 $i \leqslant n-m-1$ 时,$t=i+m$.

例 6.7.9 用算法 6.7.8 求解带宽为 3 的方程组

$$\begin{pmatrix} 5 & 6 & & \\ 6 & 5 & 6 & \\ & 6 & 5 & 6 \\ & & 6 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 11 \\ 17 \\ 17 \\ 11 \end{pmatrix}.$$

解 这里 $n=4, m=1$,数组 $c(1:4,1:2)$ 和 $\boldsymbol{b}$ 的形式如下:

$$(\boldsymbol{C} \mid \boldsymbol{b}) = \left(\begin{array}{cc:c} 0 & 5 & 11 \\ 6 & 5 & 17 \\ 6 & 5 & 17 \\ 6 & 5 & 11 \end{array}\right).$$

计算得到的 $\boldsymbol{L}$ 和 $\tilde{\boldsymbol{b}}$ 仍放在数组 $\boldsymbol{C}$ 和 $\boldsymbol{b}$ 的位置，

$$(\boldsymbol{C} \mid \boldsymbol{b}) = \left(\begin{array}{cc:c} 0 & 5 & 11 \\ 6 & -2.2 & 3.8 \\ 6 & 21.\dot{3}\dot{6} & 27.\dot{3}\dot{6} \\ 6 & 3.31489 & 3.31489 \end{array}\right).$$

利用 $\boldsymbol{L}^{\mathrm{T}}\boldsymbol{x}=\tilde{\boldsymbol{b}}$，即

$$\begin{pmatrix} 5 & 6 & & \\ & -2.2 & 6 & \\ & & 21.\dot{3}\dot{6} & 6 \\ & & & 3.31489 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 11 \\ 3.8 \\ 27.\dot{3}\dot{6} \\ 3.31489 \end{pmatrix},$$

求得解 $\boldsymbol{x}=(1,1,1,1)^{\mathrm{T}}$.

本例中的矩阵 $\boldsymbol{A}$ 对称但非负定. 说明该算法不仅适用于对称正定带状方程组，也适用于对称非奇异带状方程组.

6.8 大型稀疏方程组的直接方法

6.8.1 稀疏矩阵

大量元素为零，只有为数不多的元素不为零的矩阵，通常称之为**稀疏矩阵**(sparse matrix).

设 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$，若非零元的数目小于 $0.05n^2\sim0.1n^2$ 或为 $O(n)$，一般就认为 $\boldsymbol{A}$ 是稀疏的. 区别一个矩阵是否为稀疏矩阵，并不严格取决于非零元所占的百分比，只要是有很多零元素而且分布在

具有能够被有效利用的特点的矩阵,都可应用稀疏矩阵技术来处理. 常见的稀疏矩阵有以下几种形式,其中非零元素在实线上或阴影区域.

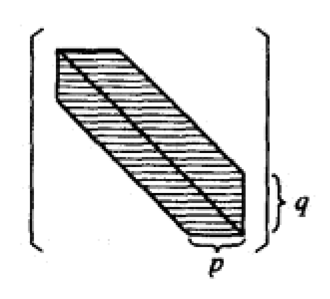


图 6.1 一般带状矩阵
带宽 $p+q+1$

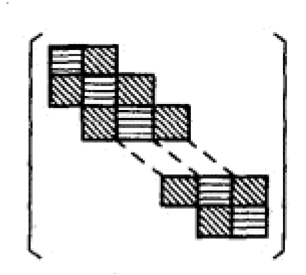


图 6.2 分块三对角阵

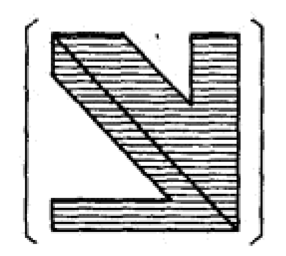


图 6.3 加边带状矩阵

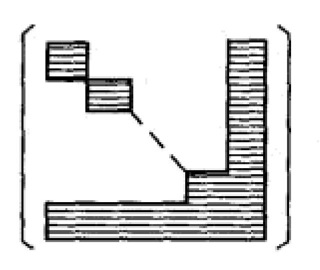


图 6.4 加边对角块矩阵

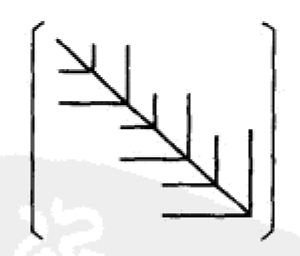


图 6.5 变带宽带状矩阵

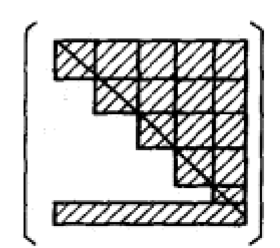


图 6.6 加边的块三角型矩阵

更一般的形式是某种绝大多数元素为零,但非零元素分布不大规则的矩阵.

具有稀疏系数矩阵的方程组称为**稀疏方程组**(sparse systems

of linear equations). 常用的解法都是针对稀疏矩阵的具体类型而提出的特殊方法,各种方法仍然依据直接法的一般原理. 如 6.7 节是针对图 6.1 的一般带状方程组和图 6.2 的分块三对角块方程组设计的解法,采用了适当的技术处理,从而大大节省了计算量和存储量.

6.8.2 高斯消去法解大型稀疏方程组所要求的形式

1. 预处理

消去法或直接三角分解法是解大型稀疏方程组的有力工具. 以消去法为例,消去过程中 $\boldsymbol{A}$ 的零元素位置上一般将出现一些新的非零元素,称为**填入**(fill-in). 希望总的填入量最小,所以,在某些情况下,在进行消元之前,需适当地重新排列方程和未知数的次序,称为对矩阵作**预处理**(precondition). 比如,设有两个系数矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$:

$$\boldsymbol{A}=\begin{bmatrix} * & * & * & * & * \\ * & * & & & \\ * & & * & & \\ * & & & * & \\ * & & & & * \end{bmatrix}, \quad \boldsymbol{B}=\begin{bmatrix} * & & & & * \\ & * & & & * \\ & & * & & * \\ & & & * & * \\ * & * & * & * & * \end{bmatrix},$$

其中 $*$ 表示非零元素. 若对系数矩阵为 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 的方程组分别用高斯法消元一步,则对应的系数矩阵形式分别为

$$\boldsymbol{A}^{(2)}=\begin{bmatrix} * & * & * & * & * \\ & * & * & * & * \\ & * & * & * & * \\ & * & * & * & * \\ & * & * & * & * \end{bmatrix}, \quad \boldsymbol{B}^{(2)}=\begin{bmatrix} * & & & & * \\ & * & & & * \\ & & * & & * \\ & & & * & * \\ & * & * & * & * \end{bmatrix},$$

这里 $\boldsymbol{A}^{(1)}=\boldsymbol{A}, \boldsymbol{B}^{(1)}=\boldsymbol{B}, \boldsymbol{A}^{(2)}$ 的右下角的 $(n-1)\times(n-1)$ 子阵可能是满矩阵,它的填入量较大,而 $\boldsymbol{B}^{(2)}$ 没有填入,消去了最左下角的

一个元素.所以事先要对 $\boldsymbol{A}$ 进行行列交换,即找排列矩阵 $\boldsymbol{P}$,使 $\boldsymbol{PAP}^{\mathrm{T}}$ 有 $\boldsymbol{B}$ 的形式.

2. 高斯消元法所需要的形式

如果有可能,对系数矩阵 $\mathbf{A}$ 作预处理成 $\hat{\mathbf{A}}$,使它有图 6.6 的形式,即

$$\hat{\mathbf{A}} = \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \cdots & \mathbf{A}_{1,p-1} & \mathbf{A}_{1p} \\ & \mathbf{A}_{22} & \cdots & \mathbf{A}_{2,p-1} & \mathbf{A}_{2p} \\ & & \ddots & \vdots & \vdots \\ & & & \mathbf{A}_{p-1,p-1} & \mathbf{A}_{p-1,p} \\ \mathbf{A}_{p1} & \mathbf{A}_{p2} & \cdots & \mathbf{A}_{p,p-1} & \mathbf{A}_{pp} \end{pmatrix},$$

其中对角线上的子矩阵 $\mathbf{A}_{ii}$,$(i=1,2,\cdots,p)$都是非奇异方阵.从对角块 $\mathbf{A}_{ii}$ 的非零元素中选取主元素.如果从 $\mathbf{A}_{11}$ 开始,依次往下进行,那么填入数仅局限在 $\hat{\mathbf{A}}$ 中那些标出的块矩阵内.如果 $\mathbf{A}$ 是对称的,$\hat{\mathbf{A}}$ 也表示成对称形式,如图 6.4 的加边对称对角块形式,这样做可节省存储量,仅需存 $\hat{\mathbf{A}}$ 的主对角线和主对角线上方的阴影部分的元素.当 $\hat{\mathbf{A}}$ 是正定矩阵时,不选主元,则消元过程能保持对称性且不必存储下三角部分.

6.8.3 压缩存储形式

一般以压缩的形式在计算机里存储大型稀疏矩阵,也就是说尽量少占用内存空间,最好不存或尽量少存系数矩阵 $\boldsymbol{A}$ 的零元素,同时给出必要的索引信息,以便方便地找到所要用的非零元素在 $\boldsymbol{A}$ 中的位置.而且当运算过程中产生新的非零元素时,有位置能方便地将它们填入.下面列出三种类型稀疏矩阵的**压缩存储形式**(packing storage scheme)及相应的稀疏方程组的特殊解法.

(1) 形如图 6.1 的等带宽带状矩阵的存储(见 6.7.3 节).将带宽为 $2m+1$ 的带状矩阵 $\mathbf{A}$ 存放在一个长方形数组 $\boldsymbol{C}(1:n,1:2m+1)$.有与之相应的带状方程组列主元消去算法 6.7.6.



(2) 形如图 6.1 的等带宽对称带状矩阵的存储(见 6.7.4 节). 将 $\boldsymbol{A}$ 的半带区存放在一个矩形数组 $\boldsymbol{C}(1:n,1:m+1)$,有与之相应的对称正定带状方程组的算法 6.7.8.

(3) 变带宽对称矩阵的存储,用两个一维数组:一个是数组 $\boldsymbol{a}(1:p)$,将 $\boldsymbol{A}$ 各行的第一个非零元至对角元(包括对角元)依次按行排列,并令 $a_{i,m_i}(i=1,2,\cdots,n)$表示 $\boldsymbol{A}$ 的各行的第一个非零元素,则

$$\boldsymbol{a}(1:p)=(a_{11},a_{2m_2},a_{22},a_{3m_3},\cdots,a_{33},\cdots,a_{im_i},\cdots,a_{ii},\cdots,a_{nm_n},\cdots,a_{nn}),$$

其中 $p=\sum_{i=1}^{n}(i-m_i+1)$ 是数组 $\boldsymbol{a}$ 中元素的个数. 另一个一维整型数组 $\boldsymbol{d}(0:n)$,其中 $d(0)=0,d(i),i=1,2,\cdots,n$,标记对角元 a_{ii} 在数组 $\boldsymbol{a}$ 中的序号.

矩阵 $\boldsymbol{A}$ 的元素 a_{ij} 处于数组 $\boldsymbol{a}$ 中的第 $d(i)-i+j$ 个位置上,即

$$a_{ij}=a(d(i)-i+j)$$

a_{i,m_i} 中的列下标 m_i 的计算公式为

$$m_i=i-(d(i)-d(i-1))+1,\quad i=1,2,\cdots,n.$$

例 6.8.1 按行压缩存储 5 阶变带宽对称矩阵,并在数组 $\boldsymbol{a}$ 中找 $\boldsymbol{A}$ 的元素 a_{42}.

解

$$\boldsymbol{A}=\begin{pmatrix} a_{11} & & & & \\ a_{21} & a_{22} & & \text{对称} & \\ & a_{32} & a_{33} & & \\ & a_{42} & 0 & a_{44} & \\ & a_{52} & a_{53} & 0 & a_{55} \end{pmatrix}.$$

$$\boldsymbol{a}=(a_{11},a_{21},a_{22},a_{32},a_{33},a_{42},0,a_{44},a_{52},a_{53},0,a_{55}),$$

$$\boldsymbol{d}=(0,1,3,5,8,12),$$

$$a_{42}=a(d(4)-4+2)=a(6).$$

这种压缩存储形式称为**包络存储**(envelope storage),由于包

络内的零元素也存储，所以对一般随机稀疏矩阵不是很节省内存空间，但判断矩阵结构较方便. 下一节的对称正定稀疏方程组的解法是针对这种包络存储设计的特殊解法.

实际使用的存储方式还有很多种，它们都是构造稀疏方程组高效算法的关键. 如逻辑尺法，链表法以及用 3 个一维数组存储随机非对称矩阵的方法等. 进一步了解可参考文献[39]等.

6.8.4 对称正定稀疏方程组的解法

对称正定稀疏方程组的解法，适用于求解大型线性方程组

$$\boldsymbol{AX} = \boldsymbol{B}, \tag{6.8.1}$$

其中 $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ 是对称正定稀疏阵，$\boldsymbol{X}, \boldsymbol{B} \in \mathbb{R}^{n \times m}$ $(m \geqslant 1)$. 也就是说，可以同时处理系数矩阵为 $\boldsymbol{A}$ 的 m 个方程组.

先对 $\boldsymbol{A}$，再对 $\boldsymbol{B}$ 分别作如下的分解：

$$\boldsymbol{A} = \boldsymbol{LDL}^{\mathrm{T}}, \tag{6.8.2}$$

$$\boldsymbol{B} = \boldsymbol{LD\widetilde{B}}, \tag{6.8.3}$$

其中 $\boldsymbol{L}$ 为单位下三角矩阵（$l_{ii}=1, i=1,2,\cdots,n, l_{ij}=0, i<j$），$\boldsymbol{D}=\mathrm{diag}(d_1, d_2, \cdots, d_n)$ 为对角矩阵. $\widetilde{\boldsymbol{B}}=(\tilde{b}_{ij})_{n\times m}$.

设矩阵 $\boldsymbol{A}$ 各行的第一个非零元素是 $a_{im_i}, i=1,2,\cdots,n$. 记 $m_{ij}=\max(m_i, m_j)$，则由式(6.8.2)和式(6.8.3)，根据矩阵乘法规则，推出矩阵 $\boldsymbol{L}, \boldsymbol{D}$ 和 $\widetilde{\boldsymbol{B}}$ 的元素的下列计算公式.

$$l_{ij} = \Big(a_{ij} - \sum_{k=m_{ij}}^{j-1} l_{ik} d_k l_{jk}\Big)\Big/ d_j, \ j = m_i, m_i+1, \cdots, i-1, i = 2,3,\cdots,n,$$

$$d_i = a_{ii} - \sum_{k=m_i}^{i-1} l_{ik}^2 d_k, \quad i = 1,2,\cdots,n,$$

$$\tilde{b}_{ik} = \Big(b_{ik} - \sum_{j=m_i}^{i-1} l_{ij} d_j \tilde{b}_{jk}\Big)\Big/ d_i, \quad i = 1,2,\cdots,n, \quad k = 1,2,\cdots,m.$$

将式(6.8.2)和式(6.8.3)代入式(6.8.1)，得到

$$L^{T}X = \widetilde{B}. \tag{6.8.4}$$

这样，方程组(6.8.1)的求解归结为方程组(6.8.4)的求解.

算法 6.8.2 对称正定稀疏方程组的解法

求解对称正定稀疏方程组 $AX=B$，其中 $A\in\mathbb{R}^{n\times n}$，$X,B\in\mathbb{R}^{n\times m}(m\geqslant 1)$. 对 A 采用 6.8.3 节(3)的变带宽压缩存储方式：用一维数组 $a(1:p)$ 和一维整型数组 $d(0:n)$ 存放对称矩阵 A，矩阵 B 存放在二维数组 $b(1:n,1:m)$ 的位置. 最终解 X 也存放在 $b(1:n,1:m)$ 的位置.

(1) 计算矩阵 L,D 和 $\widetilde{B}$. L 的元素存放在数组 $a(1:p)$ 中 A 元素的相应位置，L 的对角线元素不存. D 的元素 d_i 存放在 $a(1:p)$ 中 a_{ii} 相应的位置. $\widetilde{B}$ 存放在 $b(1:n,1:m)$ 的位置.

对于 $i=1,2,\cdots,n$

① 计算

$I=d(i)-i$； $MI\leftarrow m_i=i-(d(i)-d(i-1))+1.$

② 对于 $j=MI,MI+1,\cdots,i$

$J=d(j)-j$； $MJ\leftarrow m_j=j-(d(j)-d(j-1))+1$；

$M\leftarrow m_{ij}=\max(MI,MJ)=\max(m_i,m_j)$

$$a(I+j):=a(I+j)-\sum_{k=M}^{j-1}a(I+k)\cdot a(d(k))\cdot a(J+k)$$

如果 $j=i$，则转(3)，

否则 $a(I+j):=a(I+j)/a(d(j))$；

$b(i,k):=b(i,k)-a(I+j)\cdot a(d(j))\cdot b(j,k)$，

$k=1,2,\cdots,m$

③ 如果 $a(I+i)=0$ 则计算停止(这时 A 非正定)

否则 $b(i,k):=b(i,k)/a(d(i))$，$k=1,2,\cdots,m$

(2) 求解方程组 $L^{T}x=\widetilde{B}$.

对于 $i=n,n-1,\cdots,1$

① 计算

$I = d(i) - i$;

$MI \leftarrow m_i = i - (d(i) - d(i-1)) + 1$

② 对于 $j = MI, MI+1, \cdots, i-1$

$b(j,k) := b(j,k) - a(I+j) \cdot b(i,k), \quad k = 1,2,\cdots,m$

6.9 误差分析

6.9.1 矩阵的条件数和病态方程组

先看一个例子. 设方程组

$$\begin{pmatrix} 2 & 6 \\ 2 & 6.00001 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 8 \\ 8.00001 \end{pmatrix} \tag{6.9.1}$$

有精确解$(1,1)^T$. 对系数矩阵和方程的右端项作微小的变化,考虑扰动后的方程组

$$\begin{pmatrix} 2 & 6 \\ 2 & 5.99999 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 8 \\ 8.00002 \end{pmatrix},$$

其准确解是$(10,2)^T$. 扰动后的方程组的解面目全非了,真所谓"差之毫厘,失之千里",这种现象的出现完全是由方程组的性态决定的.

定义 6.9.1 如果方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 中,矩阵 $\boldsymbol{A}$ 和右端项 $\boldsymbol{b}$ 的变化 $\|\delta\boldsymbol{A}\|$ 和 $\|\delta\boldsymbol{b}\|$ 微小,引起解向量 $\boldsymbol{x}$ 的变化很大,则称 $\boldsymbol{A}$ 为关于解方程组和矩阵求逆的**病态矩阵**,称相应的方程组为**病态方程组**(ill-conditioned system of equations). 反之,如果 $\|\delta\boldsymbol{A}\|$ 和 $\|\delta\boldsymbol{b}\|$ 微小, $\|\delta\boldsymbol{x}\|$ 也微小,便称 $\boldsymbol{A}$ 为**良态矩阵**和 $\boldsymbol{Ax}=\boldsymbol{b}$ 为**良态方程组**(well-conditioned system of equation).

矩阵的条件数(condition number of matrix)在某种程度上能刻画方程组解对问题数据的敏感程度.

定义 6.9.2 设 $\boldsymbol{A}$ 为非奇异阵，$\|\cdot\|_p$ 为一种诱导矩阵范数，则称数

$$\mathrm{cond}_p(\boldsymbol{A}) = \|\boldsymbol{A}\|_p \|\boldsymbol{A}^{-1}\|_p$$

为矩阵的条件数.

常用的矩阵条件数为

$$\mathrm{cond}_\infty(\boldsymbol{A}) = \|\boldsymbol{A}\|_\infty \|\boldsymbol{A}^{-1}\|_\infty;$$

$$\mathrm{cond}_2(\boldsymbol{A}) = \|\boldsymbol{A}\|_2 \|\boldsymbol{A}^{-1}\|_2 = \sqrt{\frac{\lambda_{\max}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})}{\lambda_{\min}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})}};$$

$$\mathrm{cond}_1(\boldsymbol{A}) = \|\boldsymbol{A}\|_1 \|\boldsymbol{A}^{-1}\|_1.$$

其中 $\lambda_{\max}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})$ 和 $\lambda_{\min}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})$ 分别是半正定矩阵 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ 的最大和最小特征值. 称 $\mathrm{cond}_2(\boldsymbol{A})$ 为矩阵 $\boldsymbol{A}$ 的**谱条件数**(spectral condition number). 当 $\boldsymbol{A}$ 对称时，$\mathrm{cond}_2(\boldsymbol{A}) = \frac{|\lambda_{\max}(\boldsymbol{A})|}{|\lambda_{\min}(\boldsymbol{A})|}$； 当 $\boldsymbol{A}$ 对称正定时，$\mathrm{cond}_2(\boldsymbol{A}) = \frac{\lambda_{\max}(\boldsymbol{A})}{\lambda_{\min}(\boldsymbol{A})}$.

条件数有如下性质：

定理 6.9.3 设 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ 非奇异，则有

(1) $\mathrm{cond}(\boldsymbol{A}) \geqslant 1$，$\mathrm{cond}(\boldsymbol{A}) = \mathrm{cond}(\boldsymbol{A}^{-1})$；

(2) 若 $\boldsymbol{A}$ 为酉矩阵($\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A} = \boldsymbol{I}$)或正交矩阵($\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} = \boldsymbol{I}$)，则 $\mathrm{cond}_2(\boldsymbol{A}) = 1$，达到条件数的最小值；

(3) 若 $\boldsymbol{D} = \mathrm{diag}(d_1, d_2, \cdots, d_n)$ 为对角矩阵，则 $\mathrm{cond}(\boldsymbol{D}) = \max\limits_{1\leqslant i\leqslant n}|d_i| / \min\limits_{1\leqslant i\leqslant n}|d_i|$；

(4) 若 $\boldsymbol{U}$ 为正交矩阵，则 $\mathrm{cond}_2(\boldsymbol{A}) = \mathrm{cond}_2(\boldsymbol{AU}) = \mathrm{cond}_2(\boldsymbol{UA})$；

(5) $\mathrm{cond}(\alpha\boldsymbol{A}) = \mathrm{cond}(\boldsymbol{A})$，$\forall \alpha \in \mathbb{R}$，$\alpha \neq 0$.

6.9.2 方程组的敏感性

设方程组

$$\boldsymbol{Ax} = \boldsymbol{b}, \tag{6.9.2}$$

矩阵 $\boldsymbol{A}$ 和右端项 $\boldsymbol{b}$ 分别有扰动(微小变化或称误差)$\delta\boldsymbol{A}$ 和 $\delta\boldsymbol{b}$，必然引起解 $\boldsymbol{x}$ 的扰动 $\delta\boldsymbol{x}$，这时，实际得到的解是 $\boldsymbol{x}+\delta\boldsymbol{x}$，它满足扰动方程组

$$(\boldsymbol{A}+\delta\boldsymbol{A})(\boldsymbol{x}+\delta\boldsymbol{x})=\boldsymbol{b}+\delta\boldsymbol{b}.$$

定理 6.9.4 设 $\boldsymbol{A}$ 为非奇异阵，$\boldsymbol{b}\neq 0$，而且 $\|\delta\boldsymbol{A}\|$ 充分小，使 $\|\boldsymbol{A}^{-1}\|\ \|\delta\boldsymbol{A}\|<1$，则有估计式

$$\frac{\|\delta\boldsymbol{x}\|}{\|\boldsymbol{x}\|}\leqslant\frac{\operatorname{cond}(\boldsymbol{A})}{1-\|\boldsymbol{A}^{-1}\|\ \|\delta\boldsymbol{A}\|}\left(\frac{\|\delta\boldsymbol{b}\|}{\|\boldsymbol{b}\|}+\frac{\|\delta\boldsymbol{A}\|}{\|\boldsymbol{A}\|}\right).\tag{6.9.3}$$

推论 6.9.5 在定理 6.9.4 的条件下，若 $\delta\boldsymbol{A}=\boldsymbol{0}$，$\delta\boldsymbol{b}\neq\boldsymbol{0}$，则有

$$\frac{\|\delta\boldsymbol{x}\|}{\|\boldsymbol{x}\|}\leqslant\operatorname{cond}(\boldsymbol{A})\frac{\|\delta\boldsymbol{b}\|}{\|\boldsymbol{b}\|}.$$

推论 6.9.6 在定理 6.9.4 的条件下，若 $\delta\boldsymbol{b}=\boldsymbol{0}$，$\delta\boldsymbol{A}\neq\boldsymbol{0}$，则有

$$\frac{\|\delta\boldsymbol{x}\|}{\|\boldsymbol{x}\|}\leqslant\operatorname{cond}(\boldsymbol{A})\frac{\|\delta\boldsymbol{A}\|}{\|\boldsymbol{A}\|}\quad(1+O(\|\delta\boldsymbol{A}\|)).$$

由此可见，当矩阵 $\boldsymbol{A}$ 的条件数是个大数时，$\boldsymbol{b}$ 和 $\boldsymbol{A}$ 的微小扰动 $\delta\boldsymbol{b}$ 和 $\delta\boldsymbol{A}$，会引起方程组(6.9.2)的解向量有很大的扰动 $\|\delta\boldsymbol{x}\|$. 解对扰动非常敏感，也就是说当 $\operatorname{cond}(\boldsymbol{A})\gg 1$ 时，$\boldsymbol{Ax}=\boldsymbol{b}$ 是病态方程组，$\boldsymbol{A}$ 是病态的；当 $\operatorname{cond}(\boldsymbol{A})$ 相对较小时，$\boldsymbol{Ax}=\boldsymbol{b}$ 是良态方程组，$\boldsymbol{A}$ 是良态的.

例 6.9.7 希尔伯特(Hilbert)矩阵

$$\boldsymbol{H}_n=\begin{pmatrix}1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n}\\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{n+1}\\ \vdots & \vdots & \vdots & & \vdots\\ \frac{1}{n} & \frac{1}{n+1} & \frac{1}{n+2} & \cdots & \frac{1}{2n-1}\end{pmatrix}$$

是典型的病态阵. 因为 $\operatorname{cond}_2(\boldsymbol{H}_4)\approx 1.55\times 10^4$，$\operatorname{cond}_2(\boldsymbol{H}_6)\approx$

1.49×10^7，$\mathrm{cond}_2(\boldsymbol{H}_8)\approx1.53\times10^{10}$，随 n 的增加，$\boldsymbol{H}_n$ 的病态越严重. 相应的方程组 $\boldsymbol{H}_n\boldsymbol{x}=\boldsymbol{b}$ 是病态方程组.

值得注意的是：不能笼统地说某个矩阵是病态的. 因为一个矩阵可能对求解方程组和矩阵求逆是病态的，但对求特征值来说却是良态的.

利用条件数可得出另外一个重要结论. 设 $\boldsymbol{x}$ 是方程组(6.9.2)的精确解，$\tilde{\boldsymbol{x}}$为近似解，$\boldsymbol{r}=\boldsymbol{b}-\boldsymbol{A}\tilde{\boldsymbol{x}}$ 为对应于$\tilde{\boldsymbol{x}}$的剩余向量，则有如下的误差估计.

定理 6.9.8 设方程组(6.9.2)中的 $\boldsymbol{A}$ 非奇异，$\boldsymbol{b}\neq\boldsymbol{0}$，则有

$$\frac{1}{\mathrm{cond}(\boldsymbol{A})}\frac{\|\boldsymbol{r}\|}{\|\boldsymbol{b}\|}\leqslant\frac{\|\tilde{\boldsymbol{x}}-\boldsymbol{x}\|}{\|\boldsymbol{x}\|}\leqslant\mathrm{cond}(\boldsymbol{A})\frac{\|\boldsymbol{r}\|}{\|\boldsymbol{b}\|}.$$

这说明：小的剩余向量 $\boldsymbol{r}$ 并不意味着$\tilde{\boldsymbol{x}}$的高精度. 对良态方程组，小的 $\boldsymbol{r}$ 意味着$\tilde{\boldsymbol{x}}$较精确，而对病态方程组，即使 $\boldsymbol{r}$ 的范数很小，但$\tilde{\boldsymbol{x}}$仍有很大的误差.

6.9.3 病态方程组的解法

对于病态方程组，最有效的解法是采用高精度运算，如双精度、扩充精度、四倍精度以及更高的精度，以改善或减轻方程组的病态程度. 如果用无限精度运算，即不存在舍入误差，即使矩阵的条件数很大，也没有病态可言.

另一种有效的方法是在计算机运算精度受限制的情况下，也可以对病态方程组作预处理，改善方程组系数矩阵的条件数.

设有预处理矩阵 $\boldsymbol{P}$，对方程组(6.9.2)预处理后的方程组为

$$\boldsymbol{PAx}=\boldsymbol{Pb},\tag{6.9.4}$$

使

$$\mathrm{cond}(\boldsymbol{PA})\ll\mathrm{cond}(\boldsymbol{A}),\tag{6.9.5}$$

从而方程组(6.9.4)是良态的. 可选择本章的任何一种合适的方法求解.

6.9.4 舍入误差

设 $\boldsymbol{x}$ 和 $\tilde{\boldsymbol{x}}$ 分别是方程组(6.9.2)的精确解和计算解.跟踪计算过程逐步分析舍入误差来获得误差 $\|\boldsymbol{x}-\tilde{\boldsymbol{x}}\|$ 的界,称为**向前误差分析法**(forward error analysis).把计算过程舍入误差对解的影响归结为原始数据扰动对解的影响,称这种分析舍入误差的方法为**向后误差分析法**(backward error analysis).后者是目前常采用的方法.具体地说,要寻找原始数据的某种扰动 $\delta\boldsymbol{A}$ 和 $\delta\boldsymbol{b}$,使得计算解 $\tilde{\boldsymbol{x}}$ 严格地满足

$$(\boldsymbol{A}+\delta\boldsymbol{A})\tilde{\boldsymbol{x}}=\boldsymbol{b}+\delta\boldsymbol{b}.$$

找到这样的 $\delta\boldsymbol{A}$ 和 $\delta\boldsymbol{b}$ 后,便可利用式(6.9.3),估计误差 $\|\boldsymbol{x}-\tilde{\boldsymbol{x}}\|$ 的界.

已经对各种直接法作了详细地误差分析,得到了相应的 $\|\delta\boldsymbol{A}\|$ 和 $\|\delta\boldsymbol{b}\|$ 的上界.下面列出与几个主要算法有关的结果.

设 n 是矩阵 $\boldsymbol{A}$ 的阶; t 是计算机尾数的字长; a 是矩阵 $\boldsymbol{A}$ 的按模最大的元素,即 $a=\max\limits_{1\leqslant i,j\leqslant n}|a_{ij}|$.

1. 高斯消去法

$$\|\delta\boldsymbol{A}\|_\infty\leqslant c_1Ga(2n^2+n^3)2^{-t},\ \|\delta\boldsymbol{b}\|_\infty=0, \tag{6.9.6}$$

其中 $c_1\approx1$,G 为消元过程中矩阵元素的最大增长因子,即 $G=\max\limits_{\substack{1\leqslant i,j\leqslant n\\0\leqslant k\leqslant n-1}}|a_{ij}^{(k)}|/a$,$a_{ij}^{(k)}$ 是第 $k-1$ 次消元后得到的矩阵 $\boldsymbol{A}^{(k)}=(a_{ij}^{(k)})_n$ 的元素,$\boldsymbol{A}^{(1)}=\boldsymbol{A}$.

对于列主元消去法有 $G\leqslant2^{n-1}$,增长因子增长很快,达到界 2^{n-1} 的矩阵确实存在,但这种情况极少见.

对于完全选主元消去法有 $G\leqslant(n\cdot2^1\cdot3^{\frac{1}{2}}\cdot4^{\frac{1}{3}}\cdot\cdots\cdot n^{\frac{1}{n-1}})^{\frac{1}{2}}$,这时 $G\leqslant2n^{(\log n)/4+1/2}$,因此,增长因子增长缓慢.

当 $\|\boldsymbol{A}^{-1}\|\|\delta\boldsymbol{A}\|<1$ 时,将式(6.9.6)代入式(6.9.3),便得到计算解 $\tilde{\boldsymbol{x}}$ 的相对误差界

$$\frac{\| x-\tilde{x} \|_{\infty}}{\| x \|_{\infty}} \leqslant \text{cond}(\mathbf{A})\left(\frac{c_1 G}{1-\| \mathbf{A}^{-1} \|_{\infty} \| \delta \mathbf{A} \|_{\infty}}\right)(2n^2+n^3)2^{-t}. \tag{6.9.7}$$

2. 直接三角分解法.结果与高斯消去法类似

3. 对称正定矩阵的楚列斯基分解法

$$\| \delta \mathbf{A} \|_{\infty} \leqslant c_2 a(2n^2+n^3)2^{-t}, \quad \| \delta \boldsymbol{b} \|_{\infty}=0,$$

计算解 x 的相对误差界为

$$\frac{\| x-\tilde{x} \|_{\infty}}{\| x \|_{\infty}} \leqslant \text{cond}(\mathbf{A})\left(\frac{c_2}{1-\| \mathbf{A}^{-1} \|_{\infty} \| \delta \mathbf{A} \|_{\infty}}\right)(2n^2+n^3)2^{-t},$$

$$c_2 \approx 1. \tag{6.9.8}$$

从式(6.9.7)和式(6.9.8)看出,矩阵 $\mathbf{A}$ 的阶数越高、条件数越大、计算机字长越短和增长因子越大,则舍入误差对解的影响越严重.因此,直接法的计算精度取决于所选取的算法、方程组性态、所用计算机字长和矩阵的规模.当然这些估计式都是严格上界,往往是保守的.经验表明消去法有如下结果

$$\frac{\| x-\tilde{x} \|_{\infty}}{\| x \|_{\infty}} \leqslant cGn2^{-t}\text{cond}_{\infty}(\mathbf{A}).$$

7 求解线性代数方程组的迭代法

7.1 定常迭代法的基本概念

考虑线性代数方程组

$$\boldsymbol{Ax}=\boldsymbol{b},\quad \boldsymbol{A}\in\mathbb{R}^{n\times n},\quad \boldsymbol{x},\boldsymbol{b}\in\mathbb{R}^{n}. \tag{7.1.1}$$

假设 $\boldsymbol{A}$ 非奇异,方程组有惟一解 $\boldsymbol{x}^*$. 上式总可以等价地写成

$$\boldsymbol{x}=\boldsymbol{Bx}+\boldsymbol{f}. \tag{7.1.2}$$

给定初始向量 $\boldsymbol{x}^{(0)}\in\mathbb{R}^n$,可构造迭代公式

$$\boldsymbol{x}^{(k+1)}=\boldsymbol{Bx}^{(k)}+\boldsymbol{f},\quad k=0,1,2,\cdots, \tag{7.1.3}$$

其中的 $\boldsymbol{B}\in\mathbb{R}^{n\times n}$ 称为**迭代矩阵**(iteration matrix).

定义 7.1.1 如果迭代法(7.1.3)生成的序列$\{\boldsymbol{x}^{(k)}\}$满足

$$\lim_{k\to\infty}\boldsymbol{x}^{(k)}=\boldsymbol{x}^*,\quad \forall\,\boldsymbol{x}^{(0)}\in\mathbb{R}^n,$$

则称该迭代法**收敛**. 否则称为**发散**.

定理 7.1.2 迭代法(7.1.3)收敛的充分必要条件是 $\rho(\boldsymbol{B})<1$,其中 $\rho(\boldsymbol{B})$是迭代矩阵 $\boldsymbol{B}$ 的谱半径.

定理 7.1.3 设 $\boldsymbol{x}^*$ 是方程(7.1.2)的惟一解,$\|\cdot\|_v$是一种向量范数,对应的从属矩阵范数 $\|\boldsymbol{B}\|_v<1$,则由迭代式(7.1.3)产生的向量序列$\{\boldsymbol{x}^{(k)}\}$收敛且满足

$$\|\boldsymbol{x}^{(k)}-\boldsymbol{x}^*\|_v\leqslant\frac{\|\boldsymbol{B}\|_v}{1-\|\boldsymbol{B}\|_v}\|\boldsymbol{x}^{(k)}-\boldsymbol{x}^{(k-1)}\|_v,$$

$$\|\boldsymbol{x}^{(k)}-\boldsymbol{x}^*\|_v\leqslant\frac{\|\boldsymbol{B}\|_v^k}{1-\|\boldsymbol{B}\|_v}\|\boldsymbol{x}^{(1)}-\boldsymbol{x}^{(0)}\|_v.$$

虽然 $\|\boldsymbol{B}\|_v$越接近于零,序列$\{\boldsymbol{x}^{(k)}\}$收敛越快,当 $\|\boldsymbol{B}\|_v\approx 1$ 时,序列$\{\boldsymbol{x}^{(k)}\}$收敛非常慢. 总有 $\rho(\boldsymbol{B})\leqslant\|\boldsymbol{B}\|$,所以 $\rho(\boldsymbol{B})$越小,收

敛越快.

定义 7.1.4 $R_k(\boldsymbol{B})=-\ln\|\boldsymbol{B}^k\|^{\frac{1}{k}}$,称为迭代法(7.1.3)的**平均收敛率**(average convergence rate).

定义 7.1.5 $R(\boldsymbol{B})=-\ln\rho(\boldsymbol{B})$,称为迭代法(7.1.3)的**渐近收敛率**(gradual rate of convergence),或称**渐近收敛速度**.

从方程组(7.1.1)出发,可以由不同的途径得到不同的等价方程组(7.1.2),从而构成不同的迭代法(7.1.3).通常 $\boldsymbol{A}$ 可分裂为

$$\boldsymbol{A}=\boldsymbol{M}-\boldsymbol{N}, \tag{7.1.4}$$

其中 $\boldsymbol{M}$ 非奇异且 $\boldsymbol{M}^{-1}$ 易于计算,代入公式(7.1.1)得

$$\boldsymbol{x}=\boldsymbol{M}^{-1}\boldsymbol{N}\boldsymbol{x}+\boldsymbol{M}^{-1}\boldsymbol{b}. \tag{7.1.5}$$

令 $\boldsymbol{B}=\boldsymbol{M}^{-1}\boldsymbol{N}=\boldsymbol{I}-\boldsymbol{M}^{-1}\boldsymbol{A}$, $\boldsymbol{f}=\boldsymbol{M}^{-1}\boldsymbol{b}$,得迭代公式(7.1.3).$\boldsymbol{A}$ 的不同分解方式,可得到不同的迭代方法,称为求解方程组的**分裂法**(splitting method).

7.2 雅可比迭代法、JOR 法和 RF 法

设方程组

$\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$, $\boldsymbol{A}\in\mathbb{R}^{n\times n}$, $\boldsymbol{x},\boldsymbol{b}\in\mathbb{R}^n$ 且 $\boldsymbol{A}$ 非奇异.可以把 $\boldsymbol{A}$ 分解为

$$\boldsymbol{A}=\boldsymbol{D}-\boldsymbol{L}-\boldsymbol{U}, \tag{7.2.1}$$

其中 $\boldsymbol{D}$ 为对角矩阵,即 $\boldsymbol{D}=\mathrm{diag}(a_{11},a_{22},\cdots,a_{nn})$, $\boldsymbol{L}$ 和 $\boldsymbol{U}$ 分别为 $\boldsymbol{A}$ 的严格下三角部分和严格上三角部分的反号,即

$$\boldsymbol{L}=-\begin{pmatrix}0 & & & \\ a_{21} & 0 & & \\ \vdots & \vdots & \ddots & \\ a_{n1} & \cdots & a_{n,n-1} & 0\end{pmatrix},\quad \boldsymbol{U}=-\begin{pmatrix}0 & a_{12} & a_{13} & \cdots & a_{1n}\\ & 0 & a_{23} & \cdots & a_{2n}\\ & & 0 & & \vdots\\ & & & \ddots & a_{n-1,n}\\ & & & & 0\end{pmatrix}.$$

设 $\boldsymbol{D}$ 为非奇异矩阵,即 $a_{ii}\neq0$, $i=1,2,\cdots,n$.对应于一般形式的 $\boldsymbol{A}$ 的分解式(7.1.4),取 $\boldsymbol{M}=\boldsymbol{D}$, $\boldsymbol{N}=\boldsymbol{L}+\boldsymbol{U}$,代入公式(7.1.5),得

到迭代方法

$$x^{(k+1)} = D^{-1}(L+U)x^{(k)} + D^{-1}b,$$

称为**雅可比**(Jacobi)**迭代法**,简称J法.令 $B_J = D^{-1}(L+U)$,称为**雅可比迭代矩阵**.

J法的分量形式为

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left[b_i - \sum_{\substack{j=1\\j\neq i}}^{n} a_{ij}x_j^{(k)}\right], \quad k = 0,1,2,\cdots. \tag{7.2.2}$$

定理 7.2.1 J法收敛的充分必要条件是其迭代矩阵 B_J 的谱半径 $\rho(B_J)<1$.

定理 7.2.2 设 $A=A^T$,$D=\mathrm{diag}(a_{11},a_{22},\cdots,a_{nn})$,$a_{ii}>0$,$i=1,2,\cdots,n$,则J法收敛的充分必要条件是 A 及 $2D-A$ 均正定.

定理 7.2.3 若 $\|B_J\|_\nu<1$(某种向量范数导出的矩阵范数),则解 $Ax=b$ 的J法收敛.

定义 7.2.4 设 $A\in\mathbb{R}^{n\times n}$,如果存在排列矩阵 P 使得

$$PAP^{-1} = \begin{pmatrix} F & 0 \\ G & H \end{pmatrix},$$

其中 F 和 H 是方阵,0 是零矩阵,则称 A 为**可约矩阵**(reducible matrix)(或可分矩阵),否则称 A 为**不可约矩阵**(irreducible matrix)(或不可分矩阵).

定义 7.2.5 设 $A=(a_{ij})_n\in\mathbb{R}^{n\times n}$,如果

$$|a_{ii}| \geqslant \sum_{\substack{j=1\\j\neq i}}^{n} |a_{ij}|, \quad 1\leqslant i\leqslant n, \tag{7.2.3}$$

且至少对一个 i,有

$$|a_{ii}| > \sum_{\substack{j=1\\j\neq i}}^{n} |a_{ij}|, \tag{7.2.4}$$

则称 A 是**弱对角占优的**(weakly diagonally dominant).如果严格不等式(7.2.4)对 $1\leqslant i\leqslant n$ 都成立,则称 A 是**严格对角占优的**

(strictly diagonally dominant).

定理 7.2.6 设 $\boldsymbol{A}$ 为严格对角占优或不可约弱对角占优矩阵,则解 $\boldsymbol{Ax}=\boldsymbol{b}$ 的 J 法收敛.

对于实数 ω,取 $\boldsymbol{M}=\frac{1}{\omega}\boldsymbol{D}$,$\boldsymbol{N}=\frac{\boldsymbol{D}}{\omega}-\boldsymbol{A}$,代入式(7.1.5),得迭代法

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \omega\boldsymbol{D}^{-1}(\boldsymbol{Ax}^{(k)} - \boldsymbol{b}), \quad k = 0,1,2,\cdots, \tag{7.2.5}$$

称为**雅可比超松弛法**,简称 **JOR 法**(Jacobi over relaxation method).其迭代矩阵 $\boldsymbol{B}_{\mathrm{JOR}}=\boldsymbol{I}-\omega\boldsymbol{D}^{-1}\boldsymbol{A}$.

定理 7.2.7 若 $\boldsymbol{A}$ 为对称正定矩阵,则当 $2\omega^{-1}\boldsymbol{D}-\boldsymbol{A}$ 为正定矩阵时,解 $\boldsymbol{Ax}=\boldsymbol{b}$ 的 JOR 法收敛.

定理 7.2.8 设 $\boldsymbol{A}$ 为对称非奇异矩阵且其对角线元素均为正,即 $a_{ii}>0(i=1,2,\cdots,n)$,则解 $\boldsymbol{Ax}=\boldsymbol{b}$ 的 JOR 法收敛的充分必要条件是 $\boldsymbol{A}$ 与 $2\omega^{-1}\boldsymbol{D}-\boldsymbol{A}$ 均为正定.其次,$2\omega^{-1}\boldsymbol{D}-\boldsymbol{A}$ 正定等价于

$$0 \leqslant \omega < \frac{2}{1-\mu_{\min}},$$

其中 $\mu_{\min}$ 是矩阵 $\boldsymbol{B}=\boldsymbol{I}-\boldsymbol{D}^{-1}\boldsymbol{A}$ 的最小特征值.

定理 7.2.9 当解 $\boldsymbol{Ax}=\boldsymbol{b}$ 的 J 法收敛时,JOR 法对 $0<\omega\leqslant 1$ 收敛.

取 $\boldsymbol{M}=\frac{1}{\omega}\boldsymbol{I}$,$\boldsymbol{N}=\frac{1}{\omega}\boldsymbol{I}-\boldsymbol{A}$,由式(7.1.5)得到 **RF 法**(Richardson method):

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \omega(\boldsymbol{Ax}^{(k)} - \boldsymbol{b}), \quad k = 0,1,2,\cdots. \tag{7.2.6}$$

令 $\boldsymbol{B}_{\mathrm{RF}}=\boldsymbol{I}-\omega\boldsymbol{A}$,称为 **RF 法迭代矩阵**.其中实数 ω 为迭代参数.

设 $\boldsymbol{\Omega}=\mathrm{diag}(\omega_1,\omega_2,\cdots,\omega_n)$,构造如下变参数迭代法:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \boldsymbol{\Omega}(\boldsymbol{Ax}^{(k)} - \boldsymbol{b}), \quad k = 0,1,2,\cdots, \tag{7.2.7}$$

称为**广义理查森法**,简称 **GRF 法**.

定理 7.2.10 若 $\boldsymbol{A}$ 对称正定,则当 $2\boldsymbol{\Omega}^{-1}-\boldsymbol{A}$ 为正定矩阵时,解 $\boldsymbol{Ax}=\boldsymbol{b}$ 的 GRF 法收敛.

7.3 高斯-赛德尔迭代法

设方程组

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \quad \boldsymbol{A} \in \mathbb{R}^{n\times n}, \quad \boldsymbol{b},\boldsymbol{x} \in \mathbb{R}^n, \tag{7.3.1}$$

其中 $\boldsymbol{A}$ 非奇异且有分解式(7.2.1),$a_{ii}\neq 0, i=1,2,\cdots,n$.

对应于 $\boldsymbol{A}$ 的一般分解式(7.1.4),取 $\boldsymbol{M}=\boldsymbol{D}-\boldsymbol{L}, \boldsymbol{N}=\boldsymbol{U}$,代入公式(7.1.5),得迭代法

$$\boldsymbol{x}^{(k+1)} = (\boldsymbol{D}-\boldsymbol{L})^{-1}\boldsymbol{U}\boldsymbol{x}^{(k)} + (\boldsymbol{D}-\boldsymbol{L})^{-1}\boldsymbol{b}, \quad k = 0,1,2,\cdots, \tag{7.3.2}$$

称为**高斯-赛德尔**(Gauss-Seidel)**迭代法**,简称 **G-S 法**. 令 $\boldsymbol{B}_{\text{G-S}} = (\boldsymbol{D}-\boldsymbol{L})^{-1}\boldsymbol{U}$,称它为 **G-S 迭代矩阵**.

G-S 法的分量形式:

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right), \quad i = 1,2,\cdots,n.$$

定理 7.3.1 G-S 法收敛的充分必要条件是其迭代矩阵 $\boldsymbol{B}_{\text{G-S}}$ 的谱半径 $\rho(\boldsymbol{B}_{\text{G-S}})<1$.

定理 7.3.2 若 $\|\boldsymbol{B}_{\text{G-S}}\|_v<1$(某种向量范数导出的矩阵范数),则解 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 的 G-S 法收敛.

定理 7.3.3 设 $\boldsymbol{A}$ 对称正定,则解 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 的 G-S 法收敛.

定理 7.3.4 设 $\boldsymbol{A}$ 是严格对角占优或不可约弱对角占优矩阵,则解 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 的 G-S 法收敛.

定理 7.3.5 设 J 法的迭代矩阵 $\boldsymbol{B}_{\text{J}}=(b_{ij})_n$ 为非负矩阵(即 $b_{ii}=0, b_{ij}\geqslant 0, 1\leqslant i,j\leqslant n$),则下列关系有一个且只有一个成立:

(1) $\rho(\boldsymbol{B}_{\text{J}})=\rho(\boldsymbol{B}_{\text{G-S}})=0$;

(2) $0<\rho(\boldsymbol{B}_{\text{G-S}})<\rho(\boldsymbol{B}_{\text{J}})<1$;

(3) $\rho(\boldsymbol{B}_J)=\rho(\boldsymbol{B}_{G\text{-}S})=1$;

(4) $1<\rho(\boldsymbol{B}_J)<\rho(\boldsymbol{B}_{G\text{-}S})$.

定理 7.3.5 说明,在定理假设条件下,J 法和 G-S 法同时收敛,同时发散.若收敛,则 G-S 法收敛速度比 J 法快.

7.4 逐次超松弛迭代法

7.4.1 逐次超松弛迭代法

逐次超松弛(successive over relaxation)**法**的缩写为 **SOR 法**. SOR 迭代法是解大型稀疏矩阵方程组的有效方法之一.求解方程组(7.1.1)的 SOR 迭代的分量形式为

$$x_i^{(k+1)}=(1-\omega)x_i^{(k)}+\omega\left(\frac{b_i}{a_{ii}}-\sum_{j=1}^{i-1}\frac{a_{ij}}{a_{ii}}x_j^{(k+1)}-\sum_{j=i+1}^{n}\frac{a_{ij}}{a_{ii}}x_j^{(k)}\right)\quad(i=1,2,\cdots,n),\tag{7.4.1}$$

称 ω 为**松弛参数**或**松弛因子**(relaxation factor),称 $0<\omega<1$ 的迭代过程(7.4.1)为**低松弛方法**(under-relaxation method).对于一些方程组,用 G-S 迭代法得不到收敛解或不收敛,但用低松弛方法却是收敛的.称 $\omega>1$ 的迭代过程(7.4.1)为**超松弛方法**(over-relaxation method),此法可以加速 G-S 迭代方法的收敛. $\omega=1$ 的迭代过程(7.4.1)就是 G-S 迭代公式.

设方程组(7.1.1)的系数矩阵 $\boldsymbol{A}$ 有分解式(7.2.1).对于实数 ω,取 $\boldsymbol{M}=\frac{1}{\omega}(\boldsymbol{D}-\omega\boldsymbol{L})$, $\boldsymbol{N}=\frac{1}{\omega}((1-\omega)\boldsymbol{D}+\omega\boldsymbol{U})$,代入式(7.1.5),得到 SOR 迭代法的矩阵形式:

$$\boldsymbol{x}^{(k+1)}=(\boldsymbol{D}-\omega\boldsymbol{L})^{-1}((1-\omega)\boldsymbol{D}+\omega\boldsymbol{U})\boldsymbol{x}^{(k)}+\omega(\boldsymbol{D}-\omega\boldsymbol{L})^{-1}\boldsymbol{b},$$

$$k=0,1,2,\cdots. \tag{7.4.2}$$

令 $\boldsymbol{B}_{\mathrm{SOR}}=(\boldsymbol{D}-\omega \boldsymbol{L})^{-1}((1-\omega)\boldsymbol{D}+\omega \boldsymbol{U})$，称 $\boldsymbol{B}_{\mathrm{SOR}}$ 为 SOR 迭代法的**迭代矩阵**. $\rho(\boldsymbol{B}_{\mathrm{SOR}})$ 为其谱半径.

定理 7.4.1 对任意的 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$，$a_{ii}\neq 0$，$i=1,2,\cdots,n$，则对所有实数 ω，有 $\rho(\boldsymbol{B}_{\mathrm{SOR}})\geqslant|\omega-1|$.

定理 7.4.2 解 $\boldsymbol{Ax}=\boldsymbol{b}$ 的 SOR 法收敛的充分必要条件是 $\rho(\boldsymbol{B}_{\mathrm{SOR}})<1$.

定理 7.4.3 设 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$，$\boldsymbol{A}$ 对称正定，且 $0<\omega<2$，则解 $\boldsymbol{Ax}=\boldsymbol{b}$ 的 SOR 法收敛.

定理 7.4.4 设 $\boldsymbol{A}$ 为严格对角占优或不可约弱对角占优矩阵，且 $0<\omega\leqslant 1$，则解 $\boldsymbol{Ax}=\boldsymbol{b}$ 的 SOR 法收敛.

定理 7.4.5 设 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$，若 $\boldsymbol{A}$ 为具有正的对角线元素的对称矩阵，且 $0<\omega<2$，则解 $\boldsymbol{Ax}=\boldsymbol{b}$ 的 SOR 法收敛当且仅当 $\boldsymbol{A}$ 是正定的.

7.4.2 性质'A'和相容次序

定义 7.4.6 如果矩阵 $\boldsymbol{A}$ 具有形式

$$\boldsymbol{A}=\begin{pmatrix} \boldsymbol{D}_1 & \boldsymbol{U}_1 & & & \\ \boldsymbol{L}_2 & \boldsymbol{D}_2 & \boldsymbol{U}_2 & & \\ & \boldsymbol{L}_3 & \ddots & \ddots & \\ & & \ddots & \boldsymbol{D}_{m-1} & \boldsymbol{U}_{m-1} \\ & & & \boldsymbol{L}_m & \boldsymbol{D}_m \end{pmatrix}, \tag{7.4.3}$$

其中 $\boldsymbol{D}_i$ 为对角线型方阵，$\boldsymbol{L}_i$，$\boldsymbol{U}_i$ 为相应阶数的长方阵，则称 $\boldsymbol{A}$ 为 **D 型分块三对角矩阵**(diagonal form of partitioned tridiagonal matrix).

例 7.4.7

$$
A_1 = \left(\begin{array}{ccc:ccc:ccc}
a_{11} & a_{12} & 0 & a_{14} & & & & & \\
a_{21} & a_{22} & a_{23} & 0 & a_{25} & & & & \\
0 & a_{32} & a_{33} & 0 & 0 & a_{36} & & & \\
\hdashline
a_{41} & 0 & 0 & a_{44} & a_{45} & 0 & a_{47} & & \\
 & a_{52} & 0 & a_{54} & a_{55} & a_{56} & 0 & a_{58} & \\
 & & a_{63} & 0 & a_{65} & a_{66} & 0 & 0 & a_{69} \\
\hdashline
 & & & a_{74} & 0 & 0 & a_{77} & a_{78} & 0 \\
 & & & & a_{85} & 0 & a_{87} & a_{88} & a_{89} \\
 & & & & & a_{96} & 0 & a_{98} & a_{99}
\end{array}\right)
$$

$$
= \begin{pmatrix} \boldsymbol{D}_1 & \boldsymbol{U}_1 & \\ \boldsymbol{L}_2 & \boldsymbol{D}_2 & \boldsymbol{U}_2 \\ & \boldsymbol{L}_3 & \boldsymbol{D}_3 \end{pmatrix},
$$

$\boldsymbol{A}_1$ 是 D 型分块三对角矩阵.

定义 7.4.8 设 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$,如果存在一个排列矩阵 $\boldsymbol{P}$ 使得

$$
\boldsymbol{PAP}^{-1} = \begin{pmatrix} \boldsymbol{D}_1 & \boldsymbol{U}_1 \\ \boldsymbol{L}_2 & \boldsymbol{D}_2 \end{pmatrix}, \tag{7.4.4}
$$

其中 $\boldsymbol{D}_1, \boldsymbol{D}_2$ 为两个对角矩阵,则称矩阵 $\boldsymbol{A}$ 具有**性质'A'**(property 'A').

性质'A'的更确切定义如下:

定义 7.4.9 设 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$,若能将前 n 个正整数所构成的集合 $W=\{1,2,\cdots,n\}$分成两个不相交的子集合 S_1, S_2(即 $S_1+S_2=W$, S_1 与 S_2 无公共元素),使得矩阵 $\boldsymbol{A}$ 对角线以外的每一个非零元素 $a_{ij}(i\neq j)$的足标对(i,j)均满足 $i\in S_1, j\in S_2$ 或 $i\in S_2, j\in S_1$,则称此矩阵具有**性质'A'**.

例 7.4.10 用定义 7.4.9 验证例 7.4.7 中的 D 型三对角矩阵 $\boldsymbol{A}_1$ 具有性质'A'.

解 因为 $\boldsymbol{A}_1$ 是 9 阶矩阵，前 9 个正整数构成集合 $W=\{1,2,3,\cdots,9\}$，把它分成两个不相交的子集 $S_1=\{1,3,5,7,9\}$，$S_2=\{2,4,6,8\}$，$S_1+S_2=W$，S_1 和 S_2 无公共元素. 对于任一个 $a_{ij}(i\neq j)$ 的足标对 (i,j) 均满足 $i\in S_1, j\in S_2$ 或 $i\in S_2, j\in S_1$，故例 7.4.7 中的矩阵 $\boldsymbol{A}_1$ 具有性质'A'.

例 7.4.11 用定义 7.4.8 验证例 7.4.7 中的 D 型三对角矩阵 $\boldsymbol{A}_1$ 具有性质'A'.

解 将 $\boldsymbol{A}_1$ 中编号属于集合 S_1 的各行（以及相应的各列）均排在前面，而将属于集合 S_2 的均排在后面，可把矩阵 $\boldsymbol{A}_1$ 变成式(7.4.4)的矩阵形式.

将 $\boldsymbol{A}_1$ 按例 7.4.10 中的次序 S_1, S_2 重新排列后得

$$
\boldsymbol{P}\boldsymbol{A}_1\boldsymbol{P}^{-1}=\left(\begin{array}{ccccc:cccc}
a_{11} & & & & & a_{12} & a_{14} & & \\
 & a_{33} & & & & a_{32} & 0 & a_{36} & \\
 & & a_{55} & & & a_{52} & a_{54} & a_{56} & a_{58} \\
 & & & a_{77} & & & a_{74} & 0 & a_{78} \\
 & & & & a_{99} & & & a_{96} & a_{98} \\
\hdashline
a_{21} & a_{23} & a_{25} & & & a_{22} & & & \\
a_{41} & 0 & a_{45} & a_{47} & & & a_{44} & & \\
 & a_{63} & a_{65} & 0 & a_{69} & & & a_{66} & \\
 & & a_{85} & a_{87} & a_{89} & & & & a_{88}
\end{array}\right)
=\begin{pmatrix}\boldsymbol{D}_1 & \boldsymbol{U}_1\\ \boldsymbol{L}_2 & \boldsymbol{D}_2\end{pmatrix},
$$

其中 $\boldsymbol{P}$ 是排列阵，$\boldsymbol{D}_1$ 和 $\boldsymbol{D}_2$ 是对角矩阵，故由定义 7.4.8 知，矩阵 $\boldsymbol{A}_1$ 具有性质'A'.

很容易验证形如式(7.4.3)的 D 型块三对角矩阵具有性质'A'. 在矩形网格上用五点差分格式逼近二阶椭圆型微分方程所得到的系数矩阵是 D 型分块三对角矩阵，它具有性质'A'.

定义 7.4.12 若能将前 n 个正整数所构成的集合 W 分成 t 个不相交的子集 $S_1, S_2, \cdots, S_t$ $\left(\text{即} \sum_{k=1}^{t} S_k = W, S_i \text{ 与 } S_j (i \neq j) \text{无公共元素}\right)$，使得矩阵 $\boldsymbol{A}$ 的任意非对角非零元素 $a_{ij} \neq 0 (i \neq j)$ 的足标对 (i, j) 满足如下条件，若 $i \in S_k$ 时 $j \in S_{k-1}$（当 $j < i$）或 $j \in S_{k+1}$（当 $j > i$），则称此矩阵具有**相容次序**(consistently ordered).

将式(7.4.3)中属于对角线子块 $\boldsymbol{D}_k$ 的行编号记为集合 S_k，显然，$\sum_{k=1}^{m} S_k = W$ 且 S_i 与 $S_j (i \neq j)$ 无公共元素. 例 7.4.7 中矩阵 $\boldsymbol{A}_1$ 内的任意非对角线非零元素 $a_{ij} \neq 0 (i \neq j)$，其足标对 (i, j) 均满足定义 7.4.12 中的条件，所以形如式(7.4.3)的矩阵具有相容次序.

定理 7.4.13 若矩阵 $\boldsymbol{A}$ 具有相容次序，则必具有性质'A'.

定理 7.4.14 若 n 阶矩阵具有性质'A'，则 $\boldsymbol{A}$ 经过行和相应的列交换后仍具有性质'A'，也即对任意排列阵 $\boldsymbol{P}$，矩阵 $\boldsymbol{P}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{P}$ 仍具有性质'A'.

定理 7.4.15 矩阵 $\boldsymbol{A}$ 具有性质'A'的充分必要条件是存在排列矩阵 $\boldsymbol{P}$，使矩阵 $\boldsymbol{P}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{P}$ 具有相容次序矩阵的形式（见式(7.4.4)）.

定理 7.4.16 若矩阵 $\boldsymbol{A}$ 具有相容次序，则行列式 $\det(\alpha\boldsymbol{E} + \alpha^{-1}\boldsymbol{F} + \beta\boldsymbol{D})$ 的值与 α 无关，其中 $\alpha \neq 0$，而 β 为任意数，$\boldsymbol{E}$ 和 $\boldsymbol{F}$ 分别是 $\boldsymbol{A}$ 的严格下三角和上三角部分，$\boldsymbol{D}$ 是对角矩阵，且 $\boldsymbol{A} = \boldsymbol{D} + \boldsymbol{E} + \boldsymbol{F}$.

7.4.3 最优松弛因子

使逐次超松弛迭代法的渐近收敛速度 $R(\boldsymbol{B}_{\mathrm{SOR}}) = -\ln\rho(\boldsymbol{B})$ 最快的松弛因子，通常称为**最优松弛因子**(optimum relaxation factor)，用 ω_{opt} 记之. 其中 $\boldsymbol{B}_{\mathrm{SOR}}$ 为逐次超松弛迭代矩阵.

对于一般矩阵，目前尚无确定最优松弛因子 ω_{opt} 的理论结果.

仅对某些特殊类型的矩阵,例如,有性质'A'的矩阵,有相容次序的矩阵.讨论最优松弛因子 ω_{opt} 的确定.形如式(7.4.3)的D型分块三对角矩阵是具有性质'A'和相容次序矩阵的特例.由于实践中最常遇到对称正定矩阵,有对称正定的D型分块三对角矩阵 ω_{opt} 的理论计算公式.

定理 7.4.17 假定方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 的系数矩阵 $\boldsymbol{A}$ 为对称正定的D型块三对角矩阵,$\boldsymbol{A}$ 有分裂 $\boldsymbol{A}=\boldsymbol{D}-\boldsymbol{L}-\boldsymbol{L}^{\mathrm{T}}$,其中 $\boldsymbol{D}=\mathrm{diag}(a_{11},a_{22},\cdots,a_{nn})$,$\boldsymbol{L}$ 为 $\boldsymbol{A}$ 的严格下三角部分反号,则

(1) $\rho(\boldsymbol{B}_{\text{G-S}})=(\rho(\boldsymbol{B}_{\mathrm{J}}))^2$;

(2) $$\omega_{opt}=\frac{2}{1+\sqrt{1-(\rho(\boldsymbol{B}_{\mathrm{J}}))^2}}. \tag{7.4.5}$$

其中 $\rho(\boldsymbol{B}_{\mathrm{J}})$ 是雅可比迭代矩阵 $\boldsymbol{B}_{\mathrm{J}}=\boldsymbol{D}^{-1}(\boldsymbol{L}+\boldsymbol{L}^{\mathrm{T}})$ 的谱半径,$\rho(\boldsymbol{B}_{\text{G-S}})$ 是G-S迭代矩阵 $\boldsymbol{B}_{\text{G-S}}=(\boldsymbol{D}-\boldsymbol{L})^{-1}\boldsymbol{L}^{\mathrm{T}}$ 的谱半径.

由于三对角矩阵是D型块三角矩阵的特例.故定理7.4.17对于对称正定的三对角矩阵也成立.

定理 7.4.18 若矩阵 $\boldsymbol{A}=\boldsymbol{I}-\boldsymbol{L}-\boldsymbol{L}^{\mathrm{T}}$($\boldsymbol{A}$ 的对角线元素为1,$-\boldsymbol{L}$为 $\boldsymbol{A}$ 的下三角部分)为对称正定矩阵,$\boldsymbol{A}$ 的特征值 $\lambda_1\geqslant\cdots\geqslant\lambda_n>0$,矩阵 $\boldsymbol{H}=-\boldsymbol{L}-\boldsymbol{L}^{\mathrm{T}}$ 的谱半径 $\rho(\boldsymbol{H})=\sigma$,则可按如下公式粗略地确定 ω_{opt}.

$$\omega_{opt}\approx\omega^*=\begin{cases}\dfrac{2}{1+\sqrt{\lambda_1\lambda_n-\sigma^2}}, & \text{当 } \lambda_n(\lambda_1-\lambda_n)>2\sigma^2;\\[2ex] \dfrac{2}{1+\sqrt{\lambda_n^2+\sigma^2}}, & \text{当 } \lambda_n(\lambda_1-\lambda_n)\leqslant 2\sigma^2.\end{cases}$$

实际计算时,λ_1 和 σ 可用其上界,λ_n 可用其下界(大于零).

定理 7.4.19 设 $\boldsymbol{A}$ 具有相容次序,其对角元构成的对角矩阵 $\boldsymbol{D}$ 非奇异,且雅可比迭代矩阵 $\boldsymbol{B}_{\mathrm{J}}$ 的特征值全为实数,$\rho(\boldsymbol{B}_{\mathrm{J}})<1$,则

(1) $$\omega_{opt}=\frac{2}{1+\sqrt{1-(\rho(\boldsymbol{B}_{\mathrm{J}}))^2}};$$

(2) $\rho(\boldsymbol{B}_{\mathrm{SOR}},\omega_{\mathrm{opt}})=\omega_{\mathrm{opt}}-1=\left(\dfrac{\rho(\boldsymbol{B}_{\mathrm{J}})}{1+\sqrt{1-(\rho(\boldsymbol{B}_{\mathrm{J}}))^2}}\right)^2$；　(7.4.6)

(3) 若 $\omega\neq\omega_{\mathrm{opt}}$，则 $\rho(\boldsymbol{B}_{\mathrm{SOR}})>\rho(\boldsymbol{B}_{\mathrm{SOR}},\omega_{\mathrm{opt}})$；

(4)
$$\rho(\boldsymbol{B}_{\mathrm{SOR}})=\begin{cases}\left[\dfrac{\omega\rho(\boldsymbol{B}_{\mathrm{J}})+\sqrt{\omega^2(\rho(\boldsymbol{B}_{\mathrm{J}}))^2-4(\omega-1)}}{2}\right]^2, & \text{当 } 0<\omega\leqslant\omega_{\mathrm{opt}},\\ \omega-1, & \text{当 } \omega_{\mathrm{opt}}\leqslant\omega<2;\end{cases}$$
(7.4.7)

(5) 当 $0<\omega<\omega_{\mathrm{opt}}$ 时，$\rho(\boldsymbol{B}_{\mathrm{SOR}})$ 是 ω 的单调下降函数，且 $\lim\limits_{\omega\to\omega_{\mathrm{opt}}-0}\dfrac{\mathrm{d}}{\mathrm{d}\omega}\rho(\boldsymbol{B}_{\mathrm{SOR}})=-\infty$. 见图 7.1.

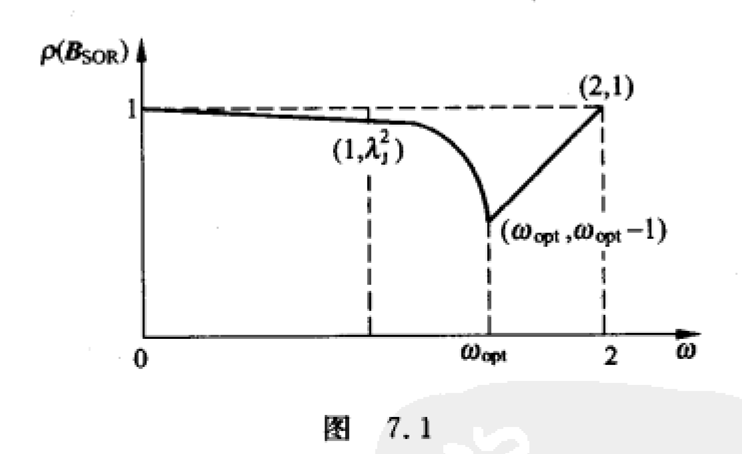


图 7.1

由图 7.1 可知，当 $\omega<\omega_{\mathrm{opt}}$ 并逐渐增加趋近于 ω_{opt} 时，$\rho(\boldsymbol{B}_{\mathrm{SOR}})$ 曲线的切线趋近于铅垂线，但当 $\omega>\omega_{\mathrm{opt}}$ 并逐渐减小趋近于 ω_{opt} 时，该切线方向不变，其斜率恒为 1.

对于大多数矩阵，目前还没有计算 ω_{opt} 的理论公式. 对于一些特殊矩阵的理论计算公式中的 $\rho(\boldsymbol{B}_{\mathrm{J}})$ 等参数也难于预先确定，还需有试算的办法. 最简单的试算确定办法：

取不同的松弛因子 $\omega_1,\omega_2,\cdots$，从同一个初始向量 $\boldsymbol{x}^{(0)}$ 出发，用逐次超松弛迭代公式(7.4.1)进行迭代，迭代次数同为 k(迭代次

数不应太少)，然后比较用 $\omega_1, \omega_2, \cdots$ 算出的剩余 $\boldsymbol{r}_{\omega_1}=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_{\omega_1}^{(k)}$，$\boldsymbol{r}_{\omega_2}=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_{\omega_2}^{(k)}, \cdots$ 或误差 $\|\boldsymbol{x}_{\omega_1}^{(k)}-\boldsymbol{x}_{\omega_1}^{(k-1)}\|, \|\boldsymbol{x}_{\omega_2}^{(k)}-\boldsymbol{x}_{\omega_2}^{(k-1)}\|, \cdots$ 其中 $\boldsymbol{x}_{\omega_i}^{(k)}$ 是用 ω_i 迭代 k 次得到的 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 的近似解. 选取达到 $\min\limits_i \|\boldsymbol{r}_{\omega_i}\|$ 或 $\min\limits_i \|\boldsymbol{x}_{\omega_i}^{(k)}-\boldsymbol{x}_{\omega_i}^{(k-1)}\|$ (即使得剩余或误差的范数(或模)最小)的松弛因子作为 ω_{opt} 的近似值. 这个方法简单而有效，特别当使用者需要多次求解具有相同系数矩阵的方程组时更是如此. 另外，如果使用者对于所求解的方程组有较深入的了解或积累了一定经验，常常可以事先定出一个包含 ω_{opt} 的不大的区间 $[\omega_a, \omega_b]$，这样可以大大减少试算次数，较快地确定 ω_{opt} 的近似值. 也可以采用优选法的原则从区间 $[\omega_a, \omega_b]$ 中选取进行试算的 ω 值，以便更快地找到 ω_{opt} 的较好近似值.

选取 ω_{opt} 自适应方法的步骤如下：

(1) 选一个 ω 使 $\omega<\omega_{\text{opt}}$，比如令 $\omega=1$；

(2) 作 n 次 SOR 迭代，用连续二次伪余量之比

$$\alpha^{(k)}=\frac{\|\Delta \boldsymbol{x}^{(k)}\|_2}{\|\Delta \boldsymbol{x}^{(k-1)}\|_2}=\frac{\left(\sum\limits_{j=1}^{n}|x_j^{(k+1)}-x_j^{(k)}|^2\right)^{1/2}}{\left(\sum\limits_{j=1}^{n}|x_j^{(k)}-x_j^{(k-1)}|^2\right)^{1/2}}$$

作为 $\rho(\boldsymbol{B}_{\text{SOR}})$ 的估值. 其中伪余量 $\Delta x^{(k)}=x^{(k+1)}-x^{(k)}=\boldsymbol{B}_{\text{SOR}}\Delta x^{(k-1)}$，当 k 相当大时，$\rho(\boldsymbol{B}_{\text{SOR}})\approx\dfrac{\|\Delta x^{(k)}\|}{\|\Delta x^{(k-1)}\|}$.

(3) 由 $\alpha^{(k)}$ 和 ω 计算相应的 $\rho(\boldsymbol{B}_{\text{J}})$.

公式(7.4.7)的第一式等价于

$$\rho(\boldsymbol{B}_{\text{SOR}})+\omega-1=\omega\rho(\boldsymbol{B}_{\text{J}})\sqrt{\rho(\boldsymbol{B}_{\text{SOR}})},$$

从而有

$$\rho(\boldsymbol{B}_{\text{J}})\approx\frac{\alpha^{(k)}+\omega-1}{\omega\sqrt{\alpha^{(k)}}}=\beta^{(k)}.$$

(4) 由 $\rho(\boldsymbol{B}_J)$的估值 $\beta^{(k)}$ 计算改进的 ω_{opt}，即由式(7.4.5)算出 ω_{opt}的第一次近似值

$$\omega_{opt}^{1}=\frac{2}{1+\sqrt{1-\beta^{(k)2}}}.$$

重复(2)～(4)可得一系列近似的 $\omega_{opt}^{(1)},\omega_{opt}^{(2)},\cdots$ 直到相邻两数之差不超过 10^{-2}(或某个适当小的正数)为止，往下可利用这个松弛因子进行迭代直到结束.

另一个近似选取 ω_{opt} 方法的步骤如下.

令

$$\delta^{(k)}=\max_{i,j}\mid x_{ij}^{(k)}-x_{ij}^{(k-1)}\mid,$$

由 $\omega=1$ 开始迭代，直到下述条件成立

$$\lg\delta^{(k-2)}-\lg\delta^{(k-1)}\approx\lg\delta^{(k-1)}-\lg\delta^{(k)}=a,$$

其中 a 是与 k 无关的常数.

然后，近似地选取最优松弛因子为

$$\omega_{opt}\approx\frac{2}{1+\sqrt{1-10^{-a}}}.$$

7.5　对称逐次超松弛迭代法

设 $\boldsymbol{Ax}=\boldsymbol{b},\boldsymbol{A}\in\mathbb{R}^{n\times n}$为非奇异矩阵，且 $\boldsymbol{A}=\boldsymbol{D}-\boldsymbol{L}-\boldsymbol{U}$(式(7.2.1))，$a_{ii}\neq0,i=1,2,\cdots,n$.

对称逐次超松弛(symmetric successive over relaxation)**法**的缩写为 **SSOR**. 解 $\boldsymbol{Ax}=\boldsymbol{b}$ 的 SSOR 法是指如下的迭代过程：假设已知第 k 次迭代的近似值 $\boldsymbol{x}^{(k)}=(x_1^{(k)},x_2^{(k)},\cdots,x_n^{(k)})^{\mathrm{T}}$，先按照自然次序($i=1,2,\cdots,n$)，用向前的逐次超松弛迭代法逐点计算 $x_i^{(k+\frac{1}{2})}$，然后按照相反的次序($i=n,n-1,\cdots,1$)，用向后的逐次超松弛法逐点计算 $x_i^{(k+1)}$，即得到 SSOR 迭代法的分量形式为

$$
\begin{cases}
x_i^{(k+\frac{1}{2})} = (1-\omega)x_i^{(k)} + \omega\left(\dfrac{b_i}{a_{ii}} - \sum\limits_{j=1}^{i-1}\dfrac{a_{ij}}{a_{ii}}x_j^{(k+\frac{1}{2})} - \sum\limits_{j=i+1}^{n}\dfrac{a_{ij}}{a_{ii}}x_j^{(k)}\right), \\
\qquad i = 1,2,\cdots,n, \\
x_i^{(k+1)} = (1-\omega)x_i^{(k+\frac{1}{2})} + \omega\left(\dfrac{b_i}{a_{ii}} - \sum\limits_{j=1}^{i-1}\dfrac{a_{ij}}{a_{ii}}x_j^{(k+\frac{1}{2})} - \sum\limits_{j=i+1}^{n}\dfrac{a_{ij}}{a_{ii}}x_j^{(k+1)}\right), \\
\qquad i = n,n-1,\cdots,1,
\end{cases}
$$

其中 ω 为松弛因子.

SSOR 迭代法的矩阵形式为

$$
\begin{cases}
\boldsymbol{x}^{(k+\frac{1}{2})} = \boldsymbol{B}_{f\text{SOR}}\boldsymbol{x}^{(k)} + \boldsymbol{f}_{f\text{SOR}}, \\
\boldsymbol{x}^{(k+1)} = \boldsymbol{B}_{b\text{SOR}}\boldsymbol{x}^{(k+\frac{1}{2})} + \boldsymbol{f}_{b\text{SOR}},
\end{cases} \tag{7.5.1}
$$

其中

$$
\begin{cases}
\boldsymbol{B}_{f\text{SOR}} = (\boldsymbol{D}-\omega\boldsymbol{L})^{-1}((1-\omega)\boldsymbol{D}+\omega\boldsymbol{U}), \\
\boldsymbol{B}_{b\text{SOR}} = (\boldsymbol{D}-\omega\boldsymbol{U})^{-1}((1-\omega)\boldsymbol{D}+\omega\boldsymbol{L}).
\end{cases}
$$

所以,SSOR 迭代矩阵是

$$
\boldsymbol{B}_{\text{SSOR}} = (\boldsymbol{D}-\omega\boldsymbol{U})^{-1}((1-\omega)\boldsymbol{D}+\omega\boldsymbol{L})(\boldsymbol{D}-\omega\boldsymbol{L})^{-1}((1-\omega)\boldsymbol{D}+\omega\boldsymbol{U}), \tag{7.5.2}
$$

$$
\boldsymbol{f}_{\text{SSOR}} = \omega(\boldsymbol{D}-\omega\boldsymbol{U})^{-1}(\boldsymbol{I}+((1-\omega)\boldsymbol{D}+\omega\boldsymbol{L})(\boldsymbol{D}-\omega\boldsymbol{L})^{-1})\boldsymbol{b}.
$$

如果将 $\boldsymbol{A}$ 分裂为 $\boldsymbol{A}=\boldsymbol{M}_{\text{SSOR}}-\boldsymbol{N}_{\text{SSOR}}$,则

$$
\boldsymbol{M}_{\text{SSOR}} = \frac{1}{\omega(2-\omega)}(\boldsymbol{D}-\omega\boldsymbol{L})\boldsymbol{D}^{-1}(\boldsymbol{D}-\omega\boldsymbol{U}), \tag{7.5.3}
$$

而且

$$
\boldsymbol{B}_{\text{SSOR}} = \boldsymbol{I} - \boldsymbol{M}_{\text{SSOR}}^{-1}\boldsymbol{A}.
$$

亦称 $\boldsymbol{M}_{\text{SSOR}}$ 为 SSOR 的预处理矩阵. 当 $\omega=1$ 时,$\boldsymbol{M}_{\text{SGS}}$ 为对称高斯-赛德尔法(简记为 SGS)预处理矩阵.

定理 7.5.1 设 $\boldsymbol{Ax}=\boldsymbol{b}$,其中 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 为对称正定矩阵,且 $0<\omega<2$,则解 $\boldsymbol{Ax}=\boldsymbol{b}$ 的 SSOR 迭代法收敛.

定理 7.5.2 设 $\boldsymbol{A}=\boldsymbol{D}-\boldsymbol{L}-\boldsymbol{U}$,$\boldsymbol{D}=\text{diag}(a_{11},a_{22},\cdots,a_{nn})$,$-\boldsymbol{L}$

和$-U$分别为A的严格下、上三角部分，如果$\rho(D^{-1}LD^{-1}U)\leqslant\frac{1}{4}$，则解$Ax=b$的SSOR迭代收敛.

对于大多数广义狄利克雷(Dirichlet)问题，只要网格节点以自然次序排列，则条件$\rho(D^{-1}LD^{-1}U)\leqslant\frac{1}{4}$满足. 如果$\rho(D^{-1}LD^{-1}U)\leqslant\frac{1}{4}$，$B_J=D^{-1}(L+U)$为相应的雅可比迭代矩阵，则可以由

$$\omega^*=\frac{2}{1+\sqrt{2(1-\rho(B_J))}}$$

计算得到较好的松弛因子，而且对于该ω^*，能够证明

$$\rho(B_{SSOR})\leqslant\frac{1-\sqrt{\frac{1-\rho(B_J)}{2}}}{1+\sqrt{\frac{1-\rho(B_J)}{2}}},$$

也可以由$\rho(D^{-1}LD^{-1}U)$的上界$\beta\left(\geqslant\frac{1}{4}\right)$和$\rho(B_J)$的上界$\alpha$，求得松弛因子$\omega^*$，即

$$\omega^*=\frac{2}{1+\sqrt{1-2\alpha+4\beta}}.$$

7.6 不完全 LU 分解

7.6.1 M 矩阵与正则分解

定义 7.6.1 设$A\in\mathbb{R}^{n\times n}$，如果$A$非奇异，$A^{-1}\geqslant 0$且$a_{ij}\leqslant 0$，$i\neq j,i,j=1,2,\cdots,n$，则称$A$为**M 矩阵**(M matrix).

定义 7.6.2 设A为$n\times n$的非奇异矩阵，$A=M-N$而且$M^{-1}\geqslant 0,N\geqslant 0$，则称$A=M-N$为$A$的一个**正则分解**(regular splitting).

$$A_{ii} = \begin{pmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ & & & -1 & 4 \end{pmatrix}_{n_i \times n_i}, \quad i = 1,2,\cdots,m.$$

易知 A 为非奇异的 M 矩阵. 将 A 分裂为 $A=A_1-A_2$，其中

$$A_1 = \begin{pmatrix} A_{11} & & & \\ & A_{22} & & \\ & & \ddots & \\ & & & A_{mm} \end{pmatrix}, \quad A_2 = \begin{pmatrix} 0 & I & & \\ I & 0 & \ddots & \\ & \ddots & \ddots & I \\ & & I & 0 \end{pmatrix},$$

由推论 7.6.3 可知，$A=A_1-A_2$ 是一个正则分解.

定理 7.6.12 方程组 $Ax=b$ 的系数矩阵 A 满足 $A^{-1}\geqslant 0$，且 $A=M-N$ 为 A 的一个正则分解，则

$\rho(M^{-1}N)=\rho(A^{-1}N)/(1+\rho(A^{-1}N))<1$，也就是说迭代法 $x^{(k+1)}=M^{-1}Nx^{(k)}+M^{-1}b$ 是收敛的.

7.6.2 不完全 LU 分解

假定方程组 $Ax=b$ 的系数矩阵 A 是大型稀疏 M 矩阵，给出 A 的一种正则分解

$$A = LU - R.$$

由于 $A\approx LU$，称这种分解为**不完全 LU 因子分解**(incomplete LU factorization). 简称 **ILU 因子分解**.

矩阵 A 的直接三角分解 $A=LU$(即杜利特尔分解)，称为**完全 LU 分解**(complete LU factorization).

定理 7.6.13 设 A 为 $n\times n$ 非奇异的 M 矩阵，则一定存在有单位下三角矩阵 L 和上三角矩阵 U 以及矩阵 R，使得

$$A = LU - R \tag{7.6.1}$$

为一正则分解，且 L 和 U 中的零元素位置可以事先任意选定，即 L

推论 7.6.3 设 $\boldsymbol{A}$ 是非奇异的 M 矩阵，而 $\boldsymbol{A}_1$ 是把 $\boldsymbol{A}$ 中某些非对角线的非零元素置为零后得到的矩阵，则 $\boldsymbol{A}=\boldsymbol{A}_1-\boldsymbol{A}_2$ 为一个正则分解.

定理 7.6.4 设 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$，$a_{ij}\leqslant 0, i\neq j, i,j=1,2,\cdots,n$，则 $\boldsymbol{A}$ 为 M 矩阵的充要条件是 $a_{ii}>0, i=1,2,\cdots,n$，且 $\boldsymbol{B}=\boldsymbol{I}-\boldsymbol{D}^{-1}\boldsymbol{A}$ 有 $\rho(\boldsymbol{B})<1$，其中 $\boldsymbol{D}=\mathrm{diag}(a_{11},a_{22},\cdots,a_{nn})$.

定理 7.6.5 n 阶矩阵 $\boldsymbol{A}$ 为 M 矩阵的充要条件为 $a_{ij}\leqslant 0, i\neq j, i,j=1,2,\cdots,n$ 且其实特征值为正.

定理 7.6.6 设 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$，$a_{ij}\leqslant 0, i\neq j, i,j=1,2,\cdots,n$，则 $\boldsymbol{A}$ 为 M 矩阵的充要条件是 $\boldsymbol{A}$ 的主子式都为正.

定理 7.6.7 设 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$，如果 $\boldsymbol{A}$ 对称正定且 $a_{ij}\leqslant 0, i\neq j, i,j=1,2,\cdots,n$，则 $\boldsymbol{A}$ 是 M 矩阵.

定理 7.6.8 设 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$，$a_{ii}>0, i=1,2,\cdots,n, a_{ij}\leqslant 0, \forall i\neq j$，且 $\boldsymbol{A}$ 为严格对角占优或不可约弱对角占优，则 $\boldsymbol{A}$ 为 M 矩阵.

定理 7.6.9 设 n 阶矩阵 $\boldsymbol{A}$ 满足 $a_{ii}>0, i=1,2,\cdots,n, a_{ij}\leqslant 0, \forall i\neq j$，且 $\boldsymbol{B}=\boldsymbol{A}+\boldsymbol{A}^{\mathrm{T}}$ 为严格对角占优或不可约弱对角占优矩阵，则 $\boldsymbol{A}$ 为 M 矩阵.

定理 7.6.10 设 n 阶矩阵 $\boldsymbol{A}$ 是 M 矩阵，$\tilde{\boldsymbol{A}}$ 是把 $\boldsymbol{A}$ 中某些非对角线上的非零元素置为零后得到的矩阵，则 $\tilde{\boldsymbol{A}}$ 也是非奇异的 M 矩阵.

例 7.6.11 设

$$\boldsymbol{A}=\begin{bmatrix}\boldsymbol{A}_{11} & -\boldsymbol{I} & & & \\ -\boldsymbol{I} & \boldsymbol{A}_{22} & -\boldsymbol{I} & & \\ & \ddots & \ddots & \ddots & \\ & & -\boldsymbol{I} & \boldsymbol{A}_{m-1,m-1} & -\boldsymbol{I} \\ & & & -\boldsymbol{I} & \boldsymbol{A}_{mm}\end{bmatrix},$$

其中

和 $\boldsymbol{U}$ 的零元素位置可由一预先给定的集合

$$\boldsymbol{P}=\{(i,j)\mid i\neq j,1\leqslant i,j\leqslant n\}$$

给出，也称 P 为**零模型**(zero pattern).

推论 7.6.14 若 $\boldsymbol{A}$ 为对称正定的 M 矩阵，$\boldsymbol{P}$ 为对称的零模型，即如果 $(i,j)\in P$，则必有 $(j,i)\in P$，则 $\boldsymbol{A}$ 有正则分解

$$\boldsymbol{A}=\boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}-\boldsymbol{R}, \tag{7.6.2}$$

其中 $\boldsymbol{L}$ 为对角线元素为正的下三角矩阵. 或有

$$\boldsymbol{A}=\boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^{\mathrm{T}}-\boldsymbol{R}, \tag{7.6.3}$$

其中 $\boldsymbol{D}$ 为对角矩阵，$\boldsymbol{L}$ 为单位下三角矩阵，剩余矩阵 $\boldsymbol{R}$ 也是对称的. $\boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$ 和 $\boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^{\mathrm{T}}$ 是正定的.

称推论 7.6.14 中的正则分解式(7.6.2)和式(7.6.3)为不完全楚列斯基分解. 这里仅给出用高斯消元法完成不完全 LU 分解式(6.2.1)的算法，算法中的零模型 $\boldsymbol{P}$ 预先给定.

算法 7.6.15 ILU 因子分解

1. 对于 $i=2,3,\cdots,n$
2. 　对于 $k=1,2,\cdots,i-1$，且若 $(i,k)\notin \boldsymbol{P}$
3. 　　$a_{ik}:=a_{ik}/a_{kk}$
4. 　　对于 $j=k+1,\cdots,n$，且对于 $(i,j)\notin \boldsymbol{P}$
5. 　　　$a_{ij}:=a_{ij}-a_{ik}*a_{kj}$

如果去掉第 2 行的"且若 $(i,k)\notin P$"和第 4 行的"且对于 $(i,j)\notin P$"，则此算法就实现了完全 LU 分解.

ILU(0)表示无填充的 ILU 因子分解.

令 $NZ(\boldsymbol{A})=\{(i,j)\mid a_{ij}\neq 0,i\neq j\}$ 表示除对角线元素之外的 $\boldsymbol{A}$ 的非零元素的指标集.

定义 7.6.16 将矩阵 $\boldsymbol{A}$ 作不完全 LU 因子分解式(7.6.1)，使 $\boldsymbol{A}-\boldsymbol{L}\boldsymbol{U}$ 在 $NZ(\boldsymbol{A})$ 的指标集上是零，称为 ILU(0)**因子分解**.

通常有无限多对 $\boldsymbol{L}$ 和 $\boldsymbol{U}$ 满足定义要求. 这里特取 $\boldsymbol{L}$ 和 $\boldsymbol{U}$ 分别与 $\boldsymbol{A}$ 的下三角部分和上三角部分有完全相同的非零结构. 这就定

义了标准的ILU(0)因子分解.

算法 7.6.17 ILU(0)因子分解

1. 对于 $i=2,3,\cdots,n$
2. 对于 $k=1,2,\cdots,i-1$ 且如果 $(i,k)\in NZ(\mathbf{A})$
3. 计算 $a_{ik}:=a_{ik}/a_{kk}$
4. 对于 $j=k+1,\cdots,n$ 且对于 $(i,j)\in NZ(\mathbf{A})$
5. 计算 $a_{ij}:=a_{ij}-a_{ik}a_{kj}$

例 7.6.18

$\mathbf{A}=$ (带状稀疏矩阵，非零对角线与主对角线相距 l)

取

$$NZ(\mathbf{A})\equiv\{(i,j)\,\big|\,|i-j|=1,l\}$$

或 $P\equiv\{(i,j)\,\big|\,|i-j|\neq 0,1,l\}$,

用算法 7.6.17 对 $\mathbf{A}$ 作 ILU(0)因子分解，得

$$\mathbf{A}=\mathbf{L}_1\mathbf{U}_1-\mathbf{R}_1,$$

其中 $\mathbf{L}_1$ 和 $\mathbf{U}_1$ 有如下的非零结构.

$\mathbf{L}_1=$ (下三角带状结构，宽度 l)，$\mathbf{U}_1=$ (上三角带状结构，宽度 l).

ILU因子分解与各种加速方法相结合，构成了一组求解大型稀疏方程组的成功和有效的方法. 特别地，它与广义共轭梯度法(见7.10.2节)相结合其计算效果令人十分满意，所以ILU因子分解是近代最常用的预处理技术.

考虑大型稀疏方程组 $\mathbf{Ax}=\mathbf{b}$，其中 $\mathbf{A}$ 为M矩阵，若对 $\mathbf{A}$ 作完

全的 LU 因子分解，则 $\boldsymbol{L}$ 和 $\boldsymbol{U}$ 会破坏 $\boldsymbol{A}$ 的稀疏结构，大大地增加了存储量. 而对 $\boldsymbol{A}$ 作 ILU 因子分解 $\boldsymbol{A}=\boldsymbol{LU}-\boldsymbol{R}$，构成迭代法

$$\boldsymbol{LUx}^{(k+1)} = \boldsymbol{Rx}^{(k)} + \boldsymbol{b} = \boldsymbol{g}, \quad k = 0,1,2,\cdots. \tag{7.6.4}$$

每个迭代步只需求解两个三角形方程组

$$\begin{cases} \boldsymbol{Ly} = \boldsymbol{g}, \\ \boldsymbol{Ux}^{(k+1)} = \boldsymbol{y}. \end{cases}$$

当满足定理 7.6.12 的条件时，求解 $\boldsymbol{Ax}=\boldsymbol{b}$ 的迭代法(7.6.4)是收敛的.

ILU(0)因子分解的精度还不足以产生合适的收敛率，更精确的 ILU 因子分解常常更有效和更可靠. 改进的办法是：设计 $\boldsymbol{L}$ 和 $\boldsymbol{U}$ 的零模型与 $\boldsymbol{A}$ 的零模型不相同. 或者说在高斯消元过程中允许在 $\boldsymbol{A}$ 的预先指令的零元素位置上出现非零元素，即允许有一些填充. 比如，在例 7.6.18 中从 ILU(0)因子分解得到的 $\boldsymbol{L}_1$ 和 $\boldsymbol{U}_1$，它们的乘积有如下形式

$$\boldsymbol{L}_1\boldsymbol{U}_1 = \begin{bmatrix} \ddots\ \ddots & \overbrace{\ddots\ \ddots\ \ddots}^{l} & \ddots\ \ddots & 0 \\ \ddots\ \ddots & \ddots\ \ddots & 0 & \ddots\ \ddots \\ \ddots\ \ddots & 0 & \ddots\ \ddots & \ddots\ \ddots \\ 0 & \ddots\ \ddots & \underbrace{\ddots\ \ddots\ \ddots}_{l} & \ddots\ \ddots \end{bmatrix}.$$

它比 $\boldsymbol{A}$ 矩阵增加了两条对角线. 如果对 $\boldsymbol{A}$ 作不完全 LU 因子分解时，规定 $\boldsymbol{L}$ 和 $\boldsymbol{U}$ 的零模型与矩阵 $\boldsymbol{L}_1\boldsymbol{U}_1$ 的零模型相同，而与 $\boldsymbol{A}$ 的零模型不同. 即 $P_1 \equiv \{(i,j) \,\big|\, |i-j| \neq 0,1,l-1,l\}$. 有 $NZ_1(\boldsymbol{A}) \equiv \{(i,j) \,\big|\, |i-j| = 1,l-1,l\}$. 这样用 $NZ_1(\boldsymbol{A})$ 执行 ILU(0)算法 7.6.17，就构造了 ILU(1)因子分解算法. 还可构造 ILU(s)，$s=2,3,\cdots$的算法，再采用阀技术得到相应的修正的各种算法. s 越大，ILU(s)算法越精确，相应的存储量也越大，存储量过大使算法不切合实际. 各种不同的 ILU 因子分解算法和它们的实现细节这里

不作详述.

7.7 块迭代法和交替方向迭代法

在 7.2 节到 7.5 节中的基本迭代法都是点迭代法：点 J 法、点 G-S 法、点 SOR 法和点 SSOR 法. 这些方法都属于分裂法，即公式 (7.1.5) 对应的迭代法. 在某些情况下矩阵 $\boldsymbol{A}$ 的另外一些分裂可能产生快速收敛的迭代. 比如，先将 $\boldsymbol{A}$ 分块，再将分块的 $\boldsymbol{A}$ 进行分裂，就构成块迭代法：**块雅可比(BJ)，块高斯-赛德尔(BG-S)，块逐次超松弛(BSOR)，对称块逐次超松弛(BSSOR)和交替方向迭代法.**

7.7.1 块雅可比法、块逐次超松弛迭代法和块高斯-赛德尔法

设 $\boldsymbol{Ax}=\boldsymbol{b}$，其中 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 为大型稀疏矩阵，将 $\boldsymbol{A}$ 分块

$$\boldsymbol{A}=\begin{pmatrix}\boldsymbol{A}_{11} & \boldsymbol{A}_{12} & \cdots & \boldsymbol{A}_{1m}\\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} & \cdots & \boldsymbol{A}_{2m}\\ \vdots & \vdots & & \vdots\\ \boldsymbol{A}_{m1} & \boldsymbol{A}_{m2} & \cdots & \boldsymbol{A}_{mm}\end{pmatrix}, \tag{7.7.1}$$

其中 $\boldsymbol{A}_{kk}\in\mathbb{R}^{n_k\times n_k}$，$k=1,2,\cdots,m$，且 $\sum\limits_{k=1}^{m}n_k=n$. 再将分块后的 $\boldsymbol{A}$ 分解为

$$\boldsymbol{A}=\boldsymbol{D}_{\mathrm{B}}-\boldsymbol{L}_{\mathrm{B}}-\boldsymbol{U}_{\mathrm{B}} \tag{7.7.2}$$

其中 $\boldsymbol{D}_{\mathrm{B}}=\mathrm{diag}(\boldsymbol{A}_{11},\boldsymbol{A}_{22},\cdots,\boldsymbol{A}_{mm})$，

$$\boldsymbol{L}_{\mathrm{B}}=-\begin{pmatrix}\boldsymbol{0} & & & & \\ \boldsymbol{A}_{21} & \boldsymbol{0} & & & \\ \boldsymbol{A}_{31} & \boldsymbol{A}_{32} & \ddots & & \\ \vdots & \vdots & \ddots & \boldsymbol{0} & \\ \boldsymbol{A}_{m1} & \boldsymbol{A}_{m2}\cdots & & \boldsymbol{A}_{m,m-1} & \boldsymbol{0}\end{pmatrix},\quad \boldsymbol{U}_{\mathrm{B}}=-\begin{pmatrix}\boldsymbol{0} & \boldsymbol{A}_{12} & \boldsymbol{A}_{13} & \cdots & \boldsymbol{A}_{1m}\\ & \boldsymbol{0} & \boldsymbol{A}_{23} & \cdots & \boldsymbol{A}_{2m}\\ & & \ddots & \ddots & \vdots\\ & & & \boldsymbol{0} & \boldsymbol{A}_{m-1,m}\\ & & & & \boldsymbol{0}\end{pmatrix}$$

对向量 $\boldsymbol{x}$ 和 $\boldsymbol{b}$ 作相应的划分

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix},$$

其中 $x_k, b_k \in \mathbb{R}^{n_k}, k=1,2,\cdots,m$.

按式(7.1.4)形式分裂 A，取 $M=D_B$，有 $N=L_B+U_B$，得到解 $Ax=b$ 的**块雅可比迭代法**，简称 **BJ 法**：

$$x^{(k+1)} = D_B^{-1}(L_B+U_B)x^{(k)} + D_B^{-1}b, \quad k=0,1,2,\cdots$$

令 $B_{BJ}=D_B^{-1}(L_B+U_B)=I-D_B^{-1}A$，称之为 **BJ 法迭代矩阵**.

BJ 法的计算公式为

$$A_{ii}x_i^{(k+1)} = b_i - \sum_{\substack{j=1 \\ j\neq i}}^{m} A_{ij}x_j^{(k)}, \quad i=1,2,\cdots,m, \quad k=0,1,2,\cdots \tag{7.7.3}$$

对每个迭代步 k，需要求解 m 个低阶方程组.

选取分裂矩阵 $M=\frac{1}{\omega}(D_B-\omega L_B)$，它为分块下三角矩阵，$\omega>0$，$A=M-N$，得到解 $Ax=b$ 的**块逐次超松弛迭代法**，简称 **BSOR 法**：

$$x^{(k+1)}=(D_B-\omega L_B)^{-1}((1-\omega)D_B+\omega U_B)x^{(k)}+\omega(D_B-\omega L_B)^{-1}b,$$
$$k=0,1,2,\cdots$$

令 $B_{BSOR}=(D_B-\omega L_B)^{-1}((1-\omega)D_B+\omega U_B)$，称之为 **BSOR 法迭代矩阵**.

BSOR 法的计算公式为

$$A_{ii}x_i^{(k+1)} = (1-\omega)A_{ii}x_i^{(k)} + \omega\left(b_i - \sum_{j=1}^{i-1}A_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n}A_{ij}x_j^{(k)}\right)$$
$$i=1,2,\cdots,m, \quad k=0,1,2,\cdots \tag{7.7.4}$$

其中 ω 为松弛因子. 当 $\omega=1$ 时，BSOR 法就是**块高斯-赛德尔迭代法**，简记 **BG-S 法**.

一般 n 是大数，而 n_i 相对是小的. 公式(7.7.3)和公式(7.7.4)中每个 i 是 n_i 个未知量的方程组，可以用直接方法求解.

7.7.2 块逐次超松弛迭代法的收敛性和最优松弛因子 $\omega^{(\pi)}$ 的选取

定理 7.7.1 若 $\boldsymbol{A}$ 为对称矩阵,$\boldsymbol{A}_{ii}(i=1,2,\cdots,m)$ 为正定矩阵,且 $0<\omega<2$,则块逐次超松弛法收敛的充分必要条件为 $\boldsymbol{A}$ 是正定矩阵.

点迭代法中的性质'A'和相容次序(见 7.4.2 节)可推广到块迭代法. 把式(7.7.1)中块矩阵 $\boldsymbol{A}$ 的 $m\times m$ 个子矩阵 $\boldsymbol{A}_{11},\boldsymbol{A}_{12},\cdots,\boldsymbol{A}_{mm}$ 与 $m\times m$ 矩阵 $\boldsymbol{C}$ 的元素相对应,使 $\boldsymbol{C}$ 的每个元素按如下方式取值:

$$c_{ij}=\begin{cases}0, & \text{若 } \boldsymbol{A}_{ij}=\boldsymbol{0},\\ 1, & \text{若 } \boldsymbol{A}_{ij}\neq \boldsymbol{0}.\end{cases}$$

当矩阵 $\boldsymbol{C}$ 具有性质'A'时,则称矩阵 $\boldsymbol{A}$ 具有性质 $\boldsymbol{A}^{(\pi)}$. 当矩阵 $\boldsymbol{C}$ 有相容次序时,则称块矩阵 $\boldsymbol{A}$ 具有 π 相容次序(π consistently ordered).

当对称正定阵 $\boldsymbol{A}$ 具有性质 $\boldsymbol{A}^{(\pi)}$ 和 π 相容次序时,块逐次超松弛迭代法的最优松弛因子为

$$\omega_{\mathrm{opt}}^{(\pi)}=\frac{2}{1+\sqrt{1-(\rho(\boldsymbol{B}_{\mathrm{BJ}}))^{2}}},$$

其中 $\rho(\boldsymbol{B}_{\mathrm{BJ}})$ 为矩阵 $\boldsymbol{B}_{\mathrm{BJ}}=\boldsymbol{I}-\boldsymbol{D}_{\mathrm{B}}^{-1}\boldsymbol{A}$ 的谱半径,$\boldsymbol{D}_{\mathrm{B}}=\mathrm{diag}(\boldsymbol{A}_{11},\boldsymbol{A}_{22},\cdots,\boldsymbol{A}_{mm})$. 相应 $\omega_{\mathrm{opt}}^{(\pi)}$ 的块逐次超松弛迭代矩阵 $\boldsymbol{B}_{\mathrm{BSOR}}$ 的谱半径为

$$\rho(\boldsymbol{B}_{\mathrm{BSOR}})=\omega_{\mathrm{opt}}^{(\pi)}-1.$$

若 $\boldsymbol{A}$ 是非奇异的 D 型分块三对角矩阵(7.4.2),且 $\boldsymbol{D}_{\mathrm{B}}=\mathrm{diag}(\boldsymbol{A}_{11},\boldsymbol{A}_{22},\cdots,\boldsymbol{A}_{mm})$ 也非奇异,则有

$$\rho(\boldsymbol{B}_{\mathrm{BG\text{-}S}})=(\rho(\boldsymbol{B}_{\mathrm{BJ}}))^{2},$$

其中 $\boldsymbol{B}_{\mathrm{BG\text{-}S}}$ 和 $\boldsymbol{B}_{\mathrm{BJ}}$ 分别是块 **G-S** 迭代矩阵和块雅可比迭代矩阵. 即 **BG-S** 法的渐近收敛速度是 **BJ** 方法的两倍. 同时,若 $\boldsymbol{B}_{\mathrm{BJ}}$ 的特征值 μ 都满足 $|\mathrm{Re}\mu|<1$,则存在 ω_0,使 $0<\omega\leqslant\omega_0<2$ 时,块逐次超松弛

法收敛. 如果 μ 全为实数，则对 $0<\omega<2$，$\rho(\boldsymbol{B}_{\text{BSOR}})<1$ 的充分必要条件是 $\rho(\boldsymbol{B}_{\text{BJ}})<1$，且最优松弛因子为

$$\omega_{\text{opt}} = \frac{2}{1+\sqrt{1-(\rho(\boldsymbol{B}_{\text{BJ}}))^2}},$$

相应的 $\boldsymbol{B}_{\text{BSOR}}$ 的谱半径为

$$\rho(\boldsymbol{B}_{\text{BSOR}}) = \omega_{\text{opt}} - 1.$$

7.7.3 交替方向隐式方法

交替方向隐式方法即 **ADI 法**(alternating direction implicit method). 在解由差分法所产生的线性方程组的迭代法中，ADI 法比松弛方法收敛得更快. 它可以看作是一种分块迭代法. ADI 法的形式很多，这里只叙述历史上最早的一个，即**皮斯曼-瑞奇福尔德**(Peaceman-Rachford)**方法**.

设矩阵 $\boldsymbol{A}$ 可以分解为

$$\begin{cases} \boldsymbol{A} = \boldsymbol{A}_1 + \boldsymbol{A}_2, \\ \boldsymbol{A}_1 = \boldsymbol{H} + \dfrac{1}{2}\boldsymbol{\Sigma}, \\ \boldsymbol{A}_2 = \boldsymbol{V} + \dfrac{1}{2}\boldsymbol{\Sigma}, \end{cases} \tag{7.7.5}$$

其中 $\boldsymbol{\Sigma}$ 是非负对角矩阵，$\boldsymbol{H}$ 和 $\boldsymbol{V}$ 是非对角线元素为非正的对称正定矩阵且满足可交换条件，即 $\boldsymbol{HV}=\boldsymbol{VH}$. 这说明 $\boldsymbol{A}_1$ 和 $\boldsymbol{A}_2$ 是对称正定的，且 $\boldsymbol{A}_1\boldsymbol{A}_2=\boldsymbol{A}_2\boldsymbol{A}_1$. 从而得到与方程组

$$\boldsymbol{Au} = \boldsymbol{b}$$

等价的两个方程组

$$\begin{cases} (\boldsymbol{A}_1 + \alpha\boldsymbol{I}_n)\boldsymbol{u} = \boldsymbol{b} - (\boldsymbol{A}_2 - \alpha\boldsymbol{I}_n)\boldsymbol{u}, \\ (\boldsymbol{A}_2 + \alpha\boldsymbol{I}_n)\boldsymbol{u} = \boldsymbol{b} - (\boldsymbol{A}_1 - \alpha\boldsymbol{I}_n)\boldsymbol{u}, \end{cases}$$

其中 α 为**加速收敛参数**(acceleration convergence parameter).

皮斯曼-瑞奇福尔德 ADI 迭代公式为

$$(\boldsymbol{A}_1+\alpha\boldsymbol{I}_n)\boldsymbol{u}^{(k+\frac{1}{2})}=\boldsymbol{b}-(\boldsymbol{A}_2-\alpha\boldsymbol{I}_n)\boldsymbol{u}^{(k)},\quad k=0,1,2,\cdots. \tag{7.7.6}$$

$$(\boldsymbol{A}_2+\alpha\boldsymbol{I}_n)\boldsymbol{u}^{(k+1)}=\boldsymbol{b}-(\boldsymbol{A}_1-\alpha\boldsymbol{I}_n)\boldsymbol{u}^{(k+\frac{1}{2})},\quad k=0,1,2,\cdots. \tag{7.7.7}$$

将两式合成一个式子为

$$\boldsymbol{u}^{(k+1)}=\boldsymbol{B}_{\mathrm{ADI}}\boldsymbol{u}^{(k)}+\boldsymbol{f}_{\mathrm{ADI}},$$

其中

$$\boldsymbol{B}_{\mathrm{ADI}}=(\boldsymbol{A}_2+\alpha\boldsymbol{I}_n)^{-1}(\boldsymbol{A}_1-\alpha\boldsymbol{I}_n)(\boldsymbol{A}_1+\alpha\boldsymbol{I}_n)^{-1}(\boldsymbol{A}_2-\alpha\boldsymbol{I}_n),$$

$$\boldsymbol{f}_{\mathrm{ADI}}=(\boldsymbol{A}_2+\alpha\boldsymbol{I}_n)^{-1}\{(\boldsymbol{A}_1-\alpha\boldsymbol{I}_n)(\boldsymbol{A}_1+\alpha\boldsymbol{I}_n)^{-1}+\boldsymbol{I}\}\boldsymbol{b}.$$

因此,$\boldsymbol{B}_{\mathrm{ADI}}$称为**ADI 迭代矩阵**,ADI 的收敛性取决于$\boldsymbol{B}_{\mathrm{ADI}}$的谱半径$\rho(\boldsymbol{B}_{\mathrm{ADI}})$.

定理 7.7.2 设 n 阶矩阵 $\boldsymbol{A}$ 有分解式(7.7.5),若 $\boldsymbol{A}_1$ 和 $\boldsymbol{A}_2$ 都是埃尔米特正定的,则对所有的 $\alpha>0$,有 $\rho(\boldsymbol{B}_{\mathrm{ADI}})<1$,即解 $\boldsymbol{Ax}=\boldsymbol{b}$ 的 ADI 法收敛.

设 μ 是矩阵 $\boldsymbol{A}_1$ 的特征值且 $0<a_1\leqslant\mu\leqslant d_1$,$\lambda$ 是矩阵 $\boldsymbol{A}_2$ 的特征值且 $0<a_2\leqslant\lambda\leqslant d_2$,$\chi$ 是矩阵 $\boldsymbol{B}_{\mathrm{ADI}}$的特征值,则有

$$\chi=\frac{(\mu-\alpha)(\lambda-\alpha)}{(\mu+\alpha)(\lambda+\alpha)}.$$

选取加速参数 α 使 χ 达到最小,得

$$\alpha=\sqrt{ad}, \tag{7.7.8}$$

其中 $a=\min(a_1,a_2)$,$d=\max(d_1,d_2)$.

用可变加速参数 α_k 更加有效.可变加速参数的皮斯曼-瑞奇福尔德 ADI 公式是

$$\begin{cases}(\boldsymbol{A}_1+\alpha_{k+1}\boldsymbol{I}_n)\boldsymbol{u}^{(k+\frac{1}{2})}=\boldsymbol{b}-(\boldsymbol{A}_2-\alpha_{k+1}\boldsymbol{I}_n)\boldsymbol{u}^{(k)},\\(\boldsymbol{A}_2+\alpha_{k+1}\boldsymbol{I}_n)\boldsymbol{u}^{(k+1)}=\boldsymbol{b}-(\boldsymbol{A}_1-\alpha_{k+1}\boldsymbol{I}_n)\boldsymbol{u}^{(k+\frac{1}{2})},\end{cases}$$

其中

$$\alpha_k=d\left(\frac{a}{d}\right)^{\frac{2k-1}{2k}}\quad(k=1,2,\cdots,m).$$

当 $k=1$ 时，上式就是式(7.7.8). 对于 α_k，采用循环使用的办法，即 $\alpha_1, \alpha_2, \cdots, \alpha_m; \alpha_1, \alpha_2, \cdots, \alpha_m; \cdots$

例 7.7.3 写出单位正方形域 $\bar{\Omega} := \{(x,y) \mid 0 \leqslant x, y \leqslant 1\}$ 上的边值问题

$$
\begin{cases}
-\dfrac{\partial^2 u}{\partial x^2}(x,y) - \dfrac{\partial^2 u}{\partial y^2}(x,y) + \sigma u(x,y) = f(x,y), & (x,y) \in \Omega, \\
u(x,y) = 0, & (x,y) \in \partial\Omega,
\end{cases} \tag{7.7.9}
$$

五点差分格式的 ADI 迭代公式. 其中 σ 为非负常数.

解 首先将单位正方形域作网格剖分，并令 $x_i = ih, y_j = jh$，$i,j = 0,1,\cdots,N,N+1, h = \dfrac{1}{N+1}, N \geqslant 1$ 为整数. 用网点集合 $\Omega_h \cup \partial\Omega_h$ 覆盖 $\Omega \cup \partial\Omega$.

$$
\Omega_h := \{(x_i, y_j) \mid i,j = 1,2,\cdots,N\},
$$

$$
\partial\Omega_h := \{(x_i,0),(x_i,1),(0,y_j),(1,y_j) \mid i,j = 0,1,\cdots,N+1\}.
$$

对边值问题(7.7.9)，分别用差分算子代替 x 和 y 方向的微分算子，因为边界条件是已知的，当 $(x_i, y_j) \in \partial\Omega_h$ 时，$u(x_i, y_j) = 0$. 用 u_{ij} 表示边值问题精确解 $u(x_i, y_j)$ 的近似值，得到

$$
\begin{aligned}
&[-u_{i-1,j} + 2u_{ij} + u_{i+1,j}] + \\
&[-u_{i,j-1} + 2u_{ij} - u_{i,j+1}] + [\sigma h^2 u_{ij}] \\
&= h^2 f_{ij}, \quad 1 \leqslant i,j \leqslant N,
\end{aligned} \tag{7.7.10}
$$

从而有线性方程组

$$
\begin{aligned}
&(4 + \sigma h^2) u_{ij} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} - u_{i,j+1} = h^2 f_{i,j}, \\
&1 \leqslant i,j \leqslant N.
\end{aligned} \tag{7.7.11}
$$

若令 $\boldsymbol{u} = (u_{11}, u_{12}, \cdots, u_{1N}, u_{21}, u_{22}, \cdots, u_{2N}, \cdots, u_{N1}, u_{N2}, \cdots, u_{NN})^{\mathrm{T}}$，$\boldsymbol{b} = (h^2 f_{11}, h^2 f_{12}, \cdots, h^2 f_{1N}, h^2 f_{21}, h^2 f_{22}, \cdots, h^2 f_{2N}, h^2 f_{N1}, \cdots, h^2 f_{NN})^{\mathrm{T}}$，则方程组(7.7.10)和(7.7.11)的矩阵形式分别为

$$
(\boldsymbol{H} + \boldsymbol{V} + \boldsymbol{\Sigma})\boldsymbol{u} = \boldsymbol{b}, \tag{7.7.12}
$$

$$
\boldsymbol{A}\boldsymbol{u} = \boldsymbol{b}, \tag{7.7.13}
$$

其中 $\boldsymbol{H}$,$\boldsymbol{V}$ 和 $\boldsymbol{\Sigma}$ 分别对应于式(7.7.10)左边的带方括号的三项.因此,$\boldsymbol{A}$ 有分解式(7.7.5).为简单起见,设 $N=3$,有

$$\boldsymbol{H}=\begin{bmatrix} 2 & -1 & 0 & & & & & & \\ -1 & 2 & -1 & & & & & & \\ 0 & -1 & 2 & & & & & & \\ & & & 2 & -1 & 0 & & & \\ & & & -1 & 2 & -1 & & & \\ & & & 0 & -1 & 2 & & & \\ & & & & & & 2 & -1 & 0 \\ & & & & & & -1 & 2 & -1 \\ & & & & & & 0 & -1 & 2 \end{bmatrix},$$

$$\boldsymbol{V}=\begin{bmatrix} 2 & 0 & 0 & -1 & & & & & \\ 0 & 2 & 0 & 0 & -1 & & & & \\ 0 & 0 & 2 & 0 & 0 & -1 & & & \\ -1 & 0 & 0 & 2 & 0 & 0 & -1 & & \\ & -1 & 0 & 0 & 2 & 0 & 0 & -1 & \\ & & -1 & 0 & 0 & 2 & 0 & 0 & -1 \\ & & & -1 & 0 & 0 & 2 & 0 & 0 \\ & & & & -1 & 0 & 0 & 2 & 0 \\ & & & & & -1 & 0 & 0 & 2 \end{bmatrix}.$$

$\boldsymbol{\Sigma}=\sigma h^2\boldsymbol{I}_N$.易知 $\boldsymbol{H}$,$\boldsymbol{V}$ 是对称正定矩阵且可交换 $\boldsymbol{HV}=\boldsymbol{VH}$,$\boldsymbol{A}_1$ 和 $\boldsymbol{A}_2$ 也是对称正定的且 $\boldsymbol{A}_1\boldsymbol{A}_2=\boldsymbol{A}_2\boldsymbol{A}_1$.于是求解方程组(7.7.13)的 ADI 迭代公式

$$\begin{cases} (\boldsymbol{A}_1+\alpha\boldsymbol{I}_n)\boldsymbol{u}^{(k+\frac{1}{2})}=\boldsymbol{f}-(\boldsymbol{A}_2-\alpha\boldsymbol{I}_n)\boldsymbol{u}^{(k)}, \quad k=0,1,2,\cdots & (7.7.14) \\ (\boldsymbol{A}_2+\alpha\boldsymbol{I}_n)\boldsymbol{u}^{(k+1)}=\boldsymbol{f}-(\boldsymbol{A}_1-\alpha\boldsymbol{I}_n)\boldsymbol{u}^{(k+\frac{1}{2})}, \quad k=0,1,2,\cdots & (7.7.15) \end{cases}$$

对任意 $\alpha>0$,是收敛的.

7.8 拟消去迭代法

考虑大型稀疏块三角方程组 $\boldsymbol{Ax}=\boldsymbol{b}$，设 $\boldsymbol{A}$ 为块三对角阵，

$$\boldsymbol{A}=\begin{pmatrix}\boldsymbol{B}_1 & \boldsymbol{C}_1 & & \\ \boldsymbol{A}_2 & \boldsymbol{B}_2 & \ddots & \\ & \ddots & \ddots & \boldsymbol{C}_{m-1} \\ & & \boldsymbol{A}_m & \boldsymbol{B}_m\end{pmatrix},$$

其中 $\boldsymbol{B}_i(i=1,2,\cdots,m)$ 为 n_i 阶方阵.

拟消去(pseudo elimination)**迭代法**，简记为 **PE 法**，它是块三对角方程组的一种有效的解法. 当 $\boldsymbol{A}$ 的次对角块的元素的绝对值比较小时，它比其他分裂迭代法收敛快.

将 $\boldsymbol{A}$ 作块直接三角分解，即

$$\boldsymbol{A}=\begin{pmatrix}\boldsymbol{D}_1 & & & \\ \boldsymbol{A}_2 & \boldsymbol{D}_2 & & \\ & \ddots & \ddots & \\ & & \boldsymbol{A}_m & \boldsymbol{D}_m\end{pmatrix}\begin{pmatrix}\boldsymbol{I}_1 & \boldsymbol{U}_1 & & \\ & \boldsymbol{I}_2 & \ddots & \\ & & \ddots & \boldsymbol{U}_{m-1} \\ & & & \boldsymbol{I}_m\end{pmatrix},$$

其中

$$\begin{cases}\boldsymbol{D}_1=\boldsymbol{B}_1, & \\ \boldsymbol{U}_i=\boldsymbol{D}_i^{-1}\boldsymbol{C}_i, & i=1,2,\cdots,m-1, \\ \boldsymbol{D}_i=\boldsymbol{B}_i-\boldsymbol{A}_i\boldsymbol{U}_{i-1}=\boldsymbol{B}_i-\boldsymbol{A}_i\boldsymbol{D}_{i-1}^{-1}\boldsymbol{C}_{i-1}, & i=2,3,\cdots,m,\end{cases} \tag{7.8.1}$$

称这种 $\boldsymbol{A}$ 的精确的块分解为**完全块因子分解**(complete block factorization).

在一定的条件下，公式(7.8.1)中的

$$\boldsymbol{D}_i^{-1}=(\boldsymbol{I}_i-\boldsymbol{B}_i^{-1}\boldsymbol{A}_i\boldsymbol{D}_{i-1}^{-1}\boldsymbol{C}_{i-1})^{-1}\boldsymbol{B}_i^{-1}\approx(\boldsymbol{I}_i+\boldsymbol{B}_i^{-1}\boldsymbol{A}_i\boldsymbol{D}_{i-1}^{-1}\boldsymbol{C}_{i-1})\boldsymbol{B}_i^{-1}, \tag{7.8.2}$$

将上式右端中的 $\boldsymbol{B}_i^{-1}\boldsymbol{A}_i\boldsymbol{D}_{i-1}^{-1}\boldsymbol{C}_{i-1}$ 省略，则有

$$D_i^{-1} \approx B_i^{-1}.$$

将式(7.8.2)中的 D_{i-1}^{-1} 用 B_{i-1}^{-1} 近似代替，然后再求逆，得

$$D_i \approx B_i(I_i + B_i^{-1}A_iB_{i-1}^{-1}C_{i-1})^{-1}. \tag{7.8.3}$$

令 $S_i = B_i(I_i + B_i^{-1}A_iB_{i-1}^{-1}C_{i-1})^{-1}$，这样得到 A 的一个近似块三角分解：

$$A \approx LU,$$

其中

$$L = \begin{pmatrix} S_1 & & & \\ A_2 & S_2 & & \\ & \ddots & \ddots & \\ & & A_m & S_m \end{pmatrix}, \quad U = \begin{pmatrix} I_1 & T_1 & & & \\ & I_2 & T_2 & & \\ & & \ddots & \ddots & \\ & & & I_{m-1} & T_{m-1} \\ & & & & I_m \end{pmatrix},$$

这里

$$\begin{cases} S_1 = B_1, \\ T_i = S_i^{-1}C_i, & i = 1,2,\cdots,m-1, \\ S_i = B_i(I_i + B_i^{-1}A_iB_{i-1}^{-1}C_{i-1})^{-1}, & i = 2,3,\cdots,m. \end{cases} \tag{7.8.4}$$

称这样的分解为 A 的**不完全块因子分解**(incomplete block factorization).

现取 $M=LU$，$N=M-A$，有求解 $Ax=b$ 的迭代公式

$$Mx^{(k+1)} = Nx^{(k)} + b, \quad k = 0,1,2,\cdots. \tag{7.8.5}$$

这就是 PE 方法.

迭代公式(7.8.5)的每次迭代，只需求解两个块三角形方程组

$$\begin{cases} Ly = Nx^{(k)} + b, \\ Ux^{(k+1)} = y. \end{cases}$$

当 A 的主对角块以外的元素比较小时，可简化不完全块因子分解. 取消式(7.8.3)中的 $B_i^{-1}A_iB_{i-1}^{-1}C_{i-1}$，则由式(7.8.3)得到 $D_i \approx B_i$. 这样，公式(7.8.4)可简化为

$$\begin{cases} \boldsymbol{S}_i = \boldsymbol{B}_i, & i = 1,2,\cdots,m, \\ \boldsymbol{T}_i = \boldsymbol{B}_i^{-1}\boldsymbol{C}_i, & i = 1,2,\cdots,m-1. \end{cases}$$

从而有 $\boldsymbol{A}$ 的另一个近似块三角分解

$$\boldsymbol{A} \approx \boldsymbol{L}_1\boldsymbol{U}_1,$$

$$\boldsymbol{L}_1 = \begin{pmatrix} \boldsymbol{B}_1 & & & \\ \boldsymbol{A}_2 & \boldsymbol{B}_2 & & \\ & \ddots & \ddots & \\ & & \boldsymbol{A}_m & \boldsymbol{B}_m \end{pmatrix}, \quad \boldsymbol{U}_1 = \begin{pmatrix} \boldsymbol{I}_1 & \boldsymbol{B}_1^{-1}\boldsymbol{C}_1 & & & \\ & \boldsymbol{I}_2 & \boldsymbol{B}_2^{-1}\boldsymbol{C}_2 & & \\ & & \ddots & \ddots & \\ & & & \boldsymbol{I}_{m-1} & \boldsymbol{B}_{m-1}^{-1}\boldsymbol{C}_{m-1} \\ & & & & \boldsymbol{I}_m \end{pmatrix},$$

这也是 $\boldsymbol{A}$ 的不完全块因子分解.

取 $\boldsymbol{M}=\boldsymbol{L}_1\boldsymbol{U}_1$, $\boldsymbol{N}=\boldsymbol{M}-\boldsymbol{A}$,得 PE 迭代公式

$$\boldsymbol{M}\boldsymbol{x}^{(k+1)} = \boldsymbol{N}\boldsymbol{x}^{(k)} + \boldsymbol{b}.$$

每次迭代也是求解两个块三对角方程组.

定理 7.8.1 若 $\boldsymbol{A}$ 为对称正定矩阵,则

(1) PE 法是可解的,即 PE 法中的逆矩阵均存在,从而算法可进行下去.

(2) PE 法收敛,即 PE 法的迭代矩阵 $\boldsymbol{B}_{PE}=\boldsymbol{M}^{-1}\boldsymbol{N}$ 的谱半径 $\rho(\boldsymbol{B}_{PE})<1$.

7.9 共轭梯度法

考虑大型稀疏方程组

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \tag{7.9.1}$$

其中 $\boldsymbol{A}$ 是 n 阶对称正定矩阵,$\boldsymbol{x},\boldsymbol{b}\in\mathbb{R}^n$. 基于变分原理的很多算法对求解大型方程组是很有效的. 共轭梯度法是其中之一.

7.9.1 变分原理

定理 7.9.1 设 $\boldsymbol{A}$ 是 n 阶对称正定矩阵,$\boldsymbol{b}\in\mathbb{R}^n$ 是已知向量,x^* 是 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 的精确解,则

$$Ax^* = b \Leftrightarrow \varphi(x^*) = \min_{x \in \mathbb{R}^n} \varphi(x), \tag{7.9.2}$$

其中

$$\varphi(x) = \frac{1}{2}(x, Ax) - (x, b). \tag{7.9.3}$$

称此定理为求解线性方程组的**变分原理**(variation principle).它将求解对称正定线性方程组的问题等价为一个 n 元二次函数(也称泛函)的极小问题.

7.9.2 最速下降法

由变分原理计算 $\varphi(x)$ 极小的算法也就给出了求解方程组(7.9.1)的算法.**最速下降法**(steepest descent method)是求二次函数 $\varphi(x)$ 的极小点 x^* 的一种迭代方法.这里从 $x^{(k)}$ 出发,沿着 $\varphi(x)$ 在 $x^{(k)}$ 点下降最快的方向搜索下一个近似点 $x^{(k+1)}$,使得 $\varphi(x^{(k+1)})$ 在该方向上达到极小值.这就是最速下降法的基本思想.

$\varphi(x)$ 在 $x^{(k)}$ 下降最快的方向是负梯度方向,即

$$-\mathbf{grad}\varphi(x)\big|_{x=x^{(k)}} = b - Ax^{(k)} = r^{(k)}.$$

取

$$x^{(k+1)} = x^{(k)} + \alpha_k r^{(k)}, \tag{7.9.4}$$

其中 α_k 待定,求 α_k 使 $x^{(k+1)}$ 为 $\varphi(x)$ 的极小值,即

$$\frac{\mathrm{d}}{\mathrm{d}\alpha_k}\varphi(x^{(k)} + \alpha_k r^{(k)}) = 0$$

得到

$$\alpha_k = \frac{(r^{(k)}, r^{(k)})}{(Ar^{(k)}, r^{(k)})},$$

从而得到最速下降法算法.

算法 7.9.2 最速下降算法

(1) 给定 $x^{(0)} \in \mathbb{R}^n$,计算 $r^{(0)} = b - Ax^{(0)}$

(2) 对于 $k=0,1,\cdots$ 直到收敛

(3) $\qquad \alpha_k=\dfrac{(\boldsymbol{r}^{(k)},\boldsymbol{r}^{(k)})}{(\boldsymbol{A}\boldsymbol{r}^{(k)},\boldsymbol{r}^{(k)})}$

(4) $\qquad \boldsymbol{x}^{(k+1)}=\boldsymbol{x}^{(k)}+\alpha_k\boldsymbol{r}^{(k)}$

(5) $\qquad \boldsymbol{r}^{(k+1)}=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}^{(k+1)}$

可以证明 $\|\boldsymbol{x}\|_A=(\boldsymbol{A}\boldsymbol{x},\boldsymbol{x})^{\frac{1}{2}}$ 是一种向量范数，称为 $\boldsymbol{A}$ 范数. 最速下降法有如下收敛定理.

定理 7.9.3 设 $\boldsymbol{A}$ 为一对称正定矩阵，λ_1,λ_n 分别为其最大、最小特征值，则对于最速下降法有

$$\|\boldsymbol{x}^{(k)}-\boldsymbol{x}^*\|_A\leqslant\left(\frac{\lambda_1-\lambda_n}{\lambda_1+\lambda_n}\right)^k\|\boldsymbol{x}_0-\boldsymbol{x}^*\|_A. \qquad (7.9.5)$$

该定理说明：对任意 $\boldsymbol{x}^{(0)}$，最速下降法收敛. 因为 $\boldsymbol{A}$ 对称正定，故 $\mathrm{cond}_2(\boldsymbol{A})=\dfrac{\lambda_1}{\lambda_n}$，当 $\boldsymbol{A}$ 的条件数大时，收敛相当慢.

因为 $\boldsymbol{r}^{(k)}$ 是 $\varphi(\boldsymbol{x})$ 在 $\boldsymbol{x}^{(k)}$ 处的最速下降方向，但这是局部的. 特别当 $\boldsymbol{r}^{(k)}$ 比较小时，实际计算中由于舍入误差的影响会偏离最速下降方向，显示数值不稳定. 所以该算法并不实用，但它的基本思想却是发展许多新算法的出发点.

7.9.3 共轭梯度法

共轭梯度法(conjugate gradient method)简称 CG 法，是求解系数矩阵对称正定的大型稀疏方程组(7.9.1)的最有效的方法之一. 它的理论基础是变分原理，即通过计算 n 元二次函数 $\varphi(\boldsymbol{x})$ 的极小点 $\boldsymbol{x}^*$，得到方程组(7.9.1)的精确解 $\boldsymbol{x}^*$. 最速下降法的搜索方向是取负梯度方向 $\boldsymbol{r}^{(k)}$，有其局部性，不是最佳选择. 共轭梯度法改进了搜索方向，即将

$$\boldsymbol{x}^{(k+1)}=\boldsymbol{x}^{(k)}+\alpha_k\boldsymbol{p}^{(k)}, \qquad (7.9.6)$$

中的搜索方向取为

$$\boldsymbol{p}^{(k)} = \boldsymbol{r}^{(k-1)} + \beta_{k-1}\boldsymbol{p}^{(k-1)}, \tag{7.9.7}$$

且 $\boldsymbol{p}^{(0)}=\boldsymbol{r}^{(0)}$.

设 $\boldsymbol{A}$ 是 n 阶对称正定矩阵. 若在$\mathbb{R}^n$中已找到一组线性无关的 $\boldsymbol{A}$ 共轭(正交)的搜索方向

$$\{\boldsymbol{p}^{(0)},\boldsymbol{p}^{(1)},\cdots,\boldsymbol{p}^{(n-1)}\},$$

其中 $\boldsymbol{p}^{(i)}\in\mathbb{R}^n, i=0,1,\cdots,n-1$. 它们可张成线性空间

$$\boldsymbol{S}_n = \mathrm{span}\{\boldsymbol{p}^{(0)},\boldsymbol{p}^{(1)},\cdots,\boldsymbol{p}^{(n-1)}\},$$

其子空间 $\boldsymbol{S}_k=\mathrm{span}\{\boldsymbol{p}^{(0)},\boldsymbol{p}^{(1)},\cdots,\boldsymbol{p}^{(k-1)}\}(k\geqslant 2)$. 迭代向量 $\boldsymbol{x}^{(k)}$ 和方程组(7.9.1)的精确解 $\boldsymbol{x}^*$ 可分别表示为

$$\boldsymbol{x}^{(k)} = \sum_{i=0}^{k-1}\alpha_i\boldsymbol{p}^{(i)},$$

$$\boldsymbol{x}^* = \sum_{i=0}^{n-1}\alpha_i\boldsymbol{p}^{(i)},$$

所以,从理论上讲,$\boldsymbol{x}^*$ 可由有限迭代步形成,至多 n 步就可得到真解 $\boldsymbol{x}^*$.

利用变分原理求 α_k,使 $\boldsymbol{x}^{(k+1)}$ 满足 $\varphi(\boldsymbol{x}^{(k+1)})=\min\limits_{x\in\mathbb{R}^n}\varphi(\boldsymbol{x})$,即 $\frac{\mathrm{d}}{\mathrm{d}\alpha_k}\varphi(\boldsymbol{x}^{(k)}+\alpha_k\boldsymbol{p}^{(k)})=0$,得

$$\alpha_k = \frac{(\boldsymbol{r}^{(k-1)},\boldsymbol{p}^{(k)})}{(\boldsymbol{A}\boldsymbol{p}^{(k)},\boldsymbol{p}^{(k)})}. \tag{7.9.8}$$

由式(7.9.7)并利用$\{\boldsymbol{p}^{(i)}\}_{i=1}^{n-1}$的 $\boldsymbol{A}$ 共轭(正交)性质($(\boldsymbol{p}^{(i)},\boldsymbol{A}\boldsymbol{p}^{(j)})=0, j\neq i$),得

$$\beta_k = -\frac{(\boldsymbol{r}^{(k)},\boldsymbol{A}\boldsymbol{p}^{(k)})}{(\boldsymbol{A}\boldsymbol{p}^{(k)},\boldsymbol{p}^{(k)})}. \tag{7.9.9}$$

易知

$$\boldsymbol{r}^{(k+1)} = \boldsymbol{r}^{(k)} - \alpha_k\boldsymbol{A}\boldsymbol{p}^{(k)}, \tag{7.9.10}$$

其中 $\boldsymbol{r}^{(k)}=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}^{(k)}$. 综合以上公式(7.9.6)～(7.9.10)构成了共

轭梯度迭代法，有以下性质定理.

定理 7.9.4 设 $\boldsymbol{A}$ 对称正定，由共轭梯度法确定的向量 $\boldsymbol{r}^{(i)}$，$\boldsymbol{p}^{(i)}$ 以及参数 α_i，β_i 满足：

(1) $(\boldsymbol{r}^{(i)},\boldsymbol{r}^{(j)})=0,\quad i\neq j$；

(2) $(\boldsymbol{p}^{(i)},\boldsymbol{A}\boldsymbol{p}^{(j)})=0,\quad i\neq j$；

(3) $(\boldsymbol{p}^{(i)},\boldsymbol{r}^{(j)})=0,\quad i<j$；

(4) $(\boldsymbol{p}^{(i)},\boldsymbol{r}^{(j)})=\|\boldsymbol{r}^{(i)}\|_2^2,\quad i\geqslant j$；

(5) $\alpha_i>0$，$\beta_i>0$.

利用以上性质和公式(7.9.10)，改写 α_k 和 β_k 的计算公式(7.9.8)和(7.9.9)，得到以下的**共轭梯度算法**.

算法 7.9.5 共轭梯度算法

(1) 给定 $\boldsymbol{x}^{(0)}=\boldsymbol{0}\in\mathbb{R}^n$，计算 $\boldsymbol{r}^{(0)}=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}^{(0)}$，取 $\boldsymbol{p}^{(1)}=\boldsymbol{r}^{(0)}$

(2) 对于 $k=1,2,\cdots$ 直到收敛

(3) $$\alpha_k:=\frac{(\boldsymbol{r}^{(k-1)},\boldsymbol{r}^{(k-1)})}{(\boldsymbol{A}\boldsymbol{p}^{(k)},\boldsymbol{p}^{(k)})}$$

(4) $$\boldsymbol{x}^{(k)}:=\boldsymbol{x}^{(k-1)}+\alpha_k\boldsymbol{p}^{(k)}$$

(5) $$\boldsymbol{r}^{(k)}:=\boldsymbol{r}^{(k-1)}-\alpha_k\boldsymbol{A}\boldsymbol{p}^{(k)}$$

(6) $$\beta_k:=\frac{(\boldsymbol{r}^{(k)},\boldsymbol{r}^{(k)})}{(\boldsymbol{r}^{(k-1)},\boldsymbol{r}^{(k-1)})}$$

(7) $$\boldsymbol{p}^{(k+1)}:=\boldsymbol{r}^{(k)}+\beta_k\boldsymbol{p}^{(k)}$$

当 $\|\boldsymbol{r}^{(k)}\|_2=0$ 或 $(\boldsymbol{p}^{(k)},\boldsymbol{A}\boldsymbol{p}^{(k)})=0$ 时，计算过程中断，这时 $\boldsymbol{x}^{(k)}$ 已等于 $\boldsymbol{x}^*$.

关于 CG 法的收敛性有以下定理.

定理 7.9.6 设 $\boldsymbol{A}$ 为对称正定矩阵，λ_1，λ_n 分别为其最大、最小特征值，则对于 CG 法有

$$\|\boldsymbol{x}^{(k)}-\boldsymbol{x}^*\|_A\leqslant 2\left(\frac{\sqrt{\lambda_1}-\sqrt{\lambda_n}}{\sqrt{\lambda_1}-\sqrt{\lambda_n}}\right)^k\|\boldsymbol{x}^{(0)}-\boldsymbol{x}^*\|_A.\tag{7.9.11}$$

对称正定阵 $\boldsymbol{A}$ 的谱条件数 $\mathrm{cond}_2(\boldsymbol{A})=\frac{\lambda_1}{\lambda_n}$，所以当它很大时，CG 法收敛很慢. 比较式(7.9.5)和式(7.9.11)可看出，CG 法比最速下降法的收敛性好得多. 可用预处理技术提高 CG 法的收敛速度并改进其数值稳定性，见 7.10 节的预处理共轭梯度法.

7.10 预处理共轭梯度法

7.10.1 预处理共轭梯度法

设方程组

$$\boldsymbol{Ax}=\boldsymbol{b},\quad \boldsymbol{x},\boldsymbol{b}\in\mathbb{R}^n, \tag{7.10.1}$$

其中 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 是一个对称正定矩阵. 由定理 7.9.6 可知，CG 法的收敛速度与 $\boldsymbol{A}$ 的条件数紧密相关，条件数愈小，收敛性愈好. 故要提高 CG 法的收敛速度，就要降低 $\boldsymbol{A}$ 的条件数，这叫做**预条件**(precondition)或叫做**预处理**. 即找一个"近似于"$\boldsymbol{A}$ 的矩阵 $\boldsymbol{M}$，它也是对称正定的. 用 $\boldsymbol{M}^{-1}$ 左乘方程组(7.10.1)两端得到其等价的方程组

$$\boldsymbol{M}^{-1}\boldsymbol{Ax}=\boldsymbol{M}^{-1}\boldsymbol{b}, \tag{7.10.2}$$

使 $\boldsymbol{M}^{-1}\boldsymbol{A}$ 的条件数比 $\boldsymbol{A}$ 的条件数小. 称 $\boldsymbol{M}$ 为**预处理器**(preconditioner). 如果 $\boldsymbol{M}^{-1}\boldsymbol{A}$ 的条件数很小，又可以用 CG 法求解方程组(7.10.2)，那么 CG 法的收敛速度可大大提高.

必须注意，即使 $\boldsymbol{A}$ 和 $\boldsymbol{M}$ 都是对称矩阵，但 $\boldsymbol{M}^{-1}\boldsymbol{A}$ 未必对称. 所以不能用 CG 法直接求解方程组(7.10.2). 为保持对称性，引入 $\boldsymbol{M}$ 内积$(\cdot,\cdot)_M$ 以及它与欧几里得内积$(\cdot,\cdot)$的关系，有如下引理.

引理 7.10.1 设 $\boldsymbol{M},\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 且均为对称正定矩阵，则

(1) $(\boldsymbol{x},\boldsymbol{y})_M=(\boldsymbol{Mx},\boldsymbol{y})=(\boldsymbol{x},\boldsymbol{My})$ (7.10.3)

是一种内积，称为 $\boldsymbol{M}$ 内积，它是自共轭的.

(2) $(M^{-1}Ax, y)_M = (Ax, y) = (x, Ay) = (x, M(M^{-1}Ay))$
$= (Mx, M^{-1}Ay) = (x, M^{-1}Ay)_M.$

$$(7.10.4)$$

(3) $(M^{-1}Ax, x)_M = (Ax, x) \geqslant 0,$ (7.10.5)

且仅当 $x=0$ 时，等号成立.

根据式(7.10.4)和式(7.10.5)，在 M 内积下 $M^{-1}A$ 是对称正定的.

定理 7.10.2 设 A 和 $M \in \mathbb{R}^{n\times n}$ 均为对称正定矩阵，$b \in \mathbb{R}^n$，则

$$M^{-1}Ax^* = M^{-1}b \Leftrightarrow \hat{\varphi}(x^*) = \min_{x\in\mathbb{R}^n} \hat{\varphi}(x),$$

其中

$$\hat{\varphi}(x) = \frac{1}{2}(M^{-1}Ax, x)_M - (M^{-1}b, x)_M.$$

由此得到 M 内积下求解方程组(7.10.2)的 CG 法.

算法 7.10.3 预处理共轭梯度法(简称 PCG 法)

(1) 计算 $r^{(0)} = b - Ax^{(0)}, z^{(0)} = M^{-1}r^{(0)}$ 和 $p^{(0)} = z^{(0)}$

(2) 对于 $k = 0, 1, 2, \cdots$ 直到收敛

(3) $\alpha_k := \dfrac{(r^{(k)}, z^{(k)})}{(Ap^{(k)}, p^{(k)})}$

(4) $x^{(k+1)} := x^{(k)} + \alpha_k p^{(k)}$

(5) $r^{(k+1)} := r^{(k)} - \alpha_k Ap^{(k)}$

(6) $z^{(k+1)} := M^{-1} r^{(k+1)}$

(7) $\beta_k := \dfrac{(r^{(k+1)}, z^{(k+1)})}{(r^{(k)}, z^{(k)})}$

(8) $p^{(k+1)} := z^{(k+1)} + \beta_k p^{(k)}$

在此算法中，每个迭代步 k 都需求解方程组

$$Mz = r, \tag{7.10.6}$$

所以要求预处理器不仅对称正定，尽可能地近似于 A，而且应使计算式(7.10.6)时，节省存储，节省时间.

7.10.2 ICCG 法

方程组(7.10.1)中的 $\boldsymbol{A}$ 是对称正定的，对 $\boldsymbol{A}$ 作不完全楚列斯基因子分解，即

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^{\mathrm{T}} - \boldsymbol{R}, \tag{7.10.7}$$

取预处理器

$$\boldsymbol{M} = \boldsymbol{L}\boldsymbol{L}^{\mathrm{T}} \approx \boldsymbol{A}, \tag{7.10.8}$$

其中 $\boldsymbol{L}$ 为下三角矩阵. 与方程组(7.10.1)等价的方程组为

$$\boldsymbol{F}\boldsymbol{y} = \boldsymbol{g}, \tag{7.10.9}$$

其中 $\boldsymbol{F}=\boldsymbol{L}^{-1}\boldsymbol{A}\boldsymbol{L}^{-\mathrm{T}}$，$\boldsymbol{y}=\boldsymbol{L}^{\mathrm{T}}\boldsymbol{x}$，$\boldsymbol{g}=\boldsymbol{L}^{-1}\boldsymbol{b}$. 因为 $\boldsymbol{A}$ 对称正定，$\boldsymbol{F}$ 亦然. 当 $\boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$ 很近似于 $\boldsymbol{A}$ 的完全分解(即 $\boldsymbol{A}=\boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$)时，$\boldsymbol{F}$ 近似于单位矩阵 $\boldsymbol{I}$，$\boldsymbol{F}$ 的条件数近似于条件数的最小值 1. 对某些不完全楚列斯基因子分解也可严格证明 $\mathrm{cond}_2(\boldsymbol{F})$ 比 $\mathrm{cond}_2(\boldsymbol{A})$ 小得多.

将 CG 算法 7.9.5 用于方程组(7.10.9)，得到不完全楚列斯基因子分解预处理 CG 算法，简称 **ICCG**(incomplete Cholesky conjugate gradient)**算法**.

算法 7.10.4 ICCG 算法

(1) 计算 $\boldsymbol{r}^{(0)}=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}^{(0)}$，$\hat{\boldsymbol{r}}^{(0)}=\boldsymbol{L}^{-1}\boldsymbol{r}^{(0)}$，$\boldsymbol{p}^{(0)}=\boldsymbol{L}^{-\mathrm{T}}\hat{\boldsymbol{r}}^{(0)}$

(2) 对于 $k=0,1,2,\cdots$ 直到收敛

(3) $\qquad \alpha_k := \dfrac{(\hat{\boldsymbol{r}}^{(k)},\hat{\boldsymbol{r}}^{(k)})}{(\boldsymbol{A}\boldsymbol{p}^{(k)},\boldsymbol{p}^{(k)})}$

(4) $\qquad \boldsymbol{x}^{(k+1)} := \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{p}^{(k)}$

(5) $\qquad \hat{\boldsymbol{r}}^{(k+1)} := \hat{\boldsymbol{r}}^{(k)} - \alpha_k \boldsymbol{L}^{-1}\boldsymbol{A}\boldsymbol{p}^{(k)}$

(6) $\qquad \beta_k := \dfrac{(\hat{\boldsymbol{r}}^{(k+1)},\hat{\boldsymbol{r}}^{(k+1)})}{(\hat{\boldsymbol{r}}^{(k)},\hat{\boldsymbol{r}}^{(k)})}$

(7) $\qquad \boldsymbol{p}^{(k+1)} := \boldsymbol{L}^{-\mathrm{T}}\hat{\boldsymbol{r}}^{(k+1)} + \beta_k \boldsymbol{p}^{(k)}$

有多种方法得到预处理器 $\boldsymbol{M}$，常用的有以下三种方法.

1. 不完全楚列斯基因子分解预处理器

最简单的不完全楚列斯基因子分解(7.10.7)是无填充的 ILL^T 因子分解,简记 $ILL^T(0)$. 类似于 ILU(0)(见算法 7.6.15),即取 $\boldsymbol{L}$ 的非零结构与 $\boldsymbol{A}$ 的下三角部分的非零结构完全一样,有如下算法.

算法 7.10.5 $ILL^T(0)$因子分解

(1) $l_{11}=(a_{11})^{\frac{1}{2}}$

(2) 对于 $i=1,2,\cdots,n$

(3) 对于 $j=1,2,\cdots,i-1$

(4) 如果 $a_{ij}=0$,则 $l_{ij}=0$,否则

(5) $l_{ij}=\left(a_{ij}-\sum\limits_{k=1}^{i-1}l_{ik}l_{kj}\right)/l_{jj}$

(6) $l_{ii}=\left(a_{ii}-\sum\limits_{k=1}^{i-1}l_{ik}\right)^{\frac{1}{2}}$

也可构造有填充的 $ILL^T(s)$,$s=1,2,\cdots$的算法,类似于 ILU(s)(见 7.6.2 节),随 s 的增加其精度也提高,直到精确的完全的楚列斯基因子分解 $\boldsymbol{A}=\boldsymbol{L}\boldsymbol{L}^T$. 当然算法的存储量和计算量也随之增加.

2. 雅可比迭代预处理器

$$\boldsymbol{M}_J=\boldsymbol{D}=\mathrm{diag}(a_{11},a_{22},\cdots,a_{nn}).$$

其中 a_{ii},$i=1,2,\cdots,n$ 为矩阵 $\boldsymbol{A}$ 的主对角线元素,称 $\boldsymbol{M}_J$ 为**雅可比迭代预处理器**,显然 $\boldsymbol{M}_J$ 对称正定.

$$\boldsymbol{L}=\boldsymbol{L}^T=\mathrm{diag}(\sqrt{a_{11}},\sqrt{a_{22}},\cdots,\sqrt{a_{nn}})$$

3. SSOR 和对称 SGS 迭代预处理器

由式(7.5.3)有 SSOR 迭代预处理器

$$\boldsymbol{M}_{SSOR}=\frac{1}{\omega(2-\omega)}(\boldsymbol{D}-\omega\boldsymbol{L}_A)\boldsymbol{D}^{-1}(\boldsymbol{D}-\omega\boldsymbol{L}_A^T),\qquad(7.10.10)$$

其中 $0<\omega<2$，L_A 为矩阵 A 的严格下三角部分反号，$D=\text{diag}(a_{11}, a_{22}, \cdots, a_{nn})$. M_{SSOR} 矩阵也是对称正定的. 由 M_{SSOR} 分解得

$$L=\frac{(D-\omega L_A)D^{-\frac{1}{2}}}{\sqrt{\omega(2-\omega)}}, \quad L^{\text{T}}=\frac{D^{-\frac{1}{2}}(D-\omega L_A^{\text{T}})}{\sqrt{\omega(2-\omega)}},$$

其中 $D^{-\frac{1}{2}}=\text{diag}\left(\frac{1}{\sqrt{a_{11}}}, \frac{1}{\sqrt{a_{22}}}, \cdots, \frac{1}{\sqrt{a_{nn}}}\right)$.

当 $\omega=1$ 时，得到对称高斯-赛德尔迭代预处理器

$$B_{\text{SGS}}=(D-L_A)D^{-1}(D-L_A^{\text{T}}) \tag{7.10.11}$$

及相应的 L 和 L^{T}.

可以证明，经 SSOR 迭代预处理器的预处理后（ω 取最优的），方程组的系数矩阵 $F=L^{-1}AL^{-\text{T}}$ 的条件数等于原方程组系数矩阵 A 的条件数的平方根，从而使 ICCG 方法加速收敛.

7.10.3 迭代预处理

在预处理 CG 算法 7.10.3 中，为了避免符号混淆，将方程组 (7.10.6) 写为

$$Qz=r, \tag{7.10.12}$$

其中 Q 对称正定，尽可能地和 A 相近且使方程组 (7.10.12) 易解. 在算法 7.10.3 的计算中可以不需要 $Q=LL^{\text{T}}$ 的分解形式.

在定常迭代中，将方程组 (7.10.1) 中的 A 分解为

$$A=M-N, \tag{7.10.13}$$

有定常迭代公式

$$x^{(k+1)}=M^{-1}Nx^{(k)}+M^{-1}b. \tag{7.10.14}$$

又有

$$A^{-1}=(I-M^{-1}N)^{-1}M^{-1},$$

令 $B=M^{-1}N$ 为式 (7.10.14) 的迭代矩阵. 当 $\rho(B)<1$ 时，得

$$A^{-1}=(I+B+B^2+\cdots)M^{-1}=\left(\sum_{k=0}^{\infty}B^k\right)M^{-1}.$$

取 $\boldsymbol{Q}\approx\boldsymbol{A}$，即

$$\boldsymbol{Q}^{-1}=(1+\boldsymbol{B}+\boldsymbol{B}^2+\cdots+\boldsymbol{B}^p)\boldsymbol{M}^{-1}, \tag{7.10.15}$$

显然 $\boldsymbol{Q}$ 是对称的，当 p 足够大时，它是正定的且 $\boldsymbol{Q}^{-1}$ 相当靠近 $\boldsymbol{A}^{-1}$. 从式(7.10.12)和式(7.10.15)得到

$$\boldsymbol{z}=\boldsymbol{Q}^{-1}\boldsymbol{r}=(\boldsymbol{I}+\boldsymbol{B}+\boldsymbol{B}^2+\cdots+\boldsymbol{B}^p)\boldsymbol{M}^{-1}\boldsymbol{r}, \tag{7.10.16}$$

如果取 $\boldsymbol{z}^{(0)}=\boldsymbol{M}^{-1}\boldsymbol{r}$，用 $\rho(\boldsymbol{B})<1$ 的定常迭代法求解

$$\boldsymbol{A}\boldsymbol{z}=\boldsymbol{r} \tag{7.10.17}$$

迭代 p 步就得到式(7.10.16)的结果.

对于求解方程组(7.10.17)的**雅可比**迭代法，$\boldsymbol{M}=\boldsymbol{D}=\mathrm{diag}(a_{11},a_{22},\cdots,a_{nn})$ 是对称正定的，$\boldsymbol{B}_J=\boldsymbol{D}^{-1}\boldsymbol{N}$ 为**雅可比**迭代矩阵. 若 $\rho(\boldsymbol{B}_J)<1$，取 $\boldsymbol{z}^{(0)}=\boldsymbol{D}^{-1}\boldsymbol{r}$，则迭代 p 步就得到 $\boldsymbol{Q}\boldsymbol{z}=\boldsymbol{r}$ 的近似解.

对称高斯-赛德尔法的 $\boldsymbol{M}=(\boldsymbol{D}-\boldsymbol{L}_A)\boldsymbol{D}^{-1}(\boldsymbol{D}-\boldsymbol{L}_A^{\mathrm{T}})$（见式(7.10.11)）也对称正定. 因为 $\boldsymbol{A}$ 对称正定，用对称高斯-赛德尔法求解方程组(7.10.17)收敛，取 $\boldsymbol{z}^{(0)}=\boldsymbol{M}^{-1}\boldsymbol{r}$，则迭代 p 步得到 $\boldsymbol{Q}\boldsymbol{z}=\boldsymbol{r}$ 的近似解. 同理也可以用 $0<\omega<2$ 的对称逐次超松弛法做共轭梯度法的迭代预处理.

无论用雅可比迭代，还是用对称高斯-赛德尔和对称逐次超松弛迭代预处理，一般只需进行几次迭代就足够了.

7.11 阿诺尔德算法

7.11.1 伽辽金原理和克雷洛夫子空间

考虑大型线性方程组

$$\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b},\quad \boldsymbol{x},\boldsymbol{b}\in\mathbb{R}^n, \tag{7.11.1}$$

其中 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 为非奇异大型稀疏矩阵且不必对称. 取 $\boldsymbol{x}_0\in\mathbb{R}^n$ 是任意向量，令

$$\boldsymbol{x}=\boldsymbol{x}_0+\boldsymbol{z}, \tag{7.11.2}$$

$r_0=b-Ax_0$,则方程组(7.11.1)等价于

$$Az = r_0. \tag{7.11.3}$$

设 K_m 和 L_m 是$\mathbb{R}^n$中两个 m 维子空间,分别由$\{v_i\}_{i=1}^m$和$\{w_i\}_{i=1}^m$张成,其中$v_i, w_i\in\mathbb{R}^n$.

求解方程组(7.11.3)的**伽辽金**(Galerkin)原理是:在 K_m 中找方程组(7.11.3)的近似解 z_m,使残余向量 r_0-Az_m 和 L_m 中的所有向量正交,即求 $z_m\in K_m$ 使

$$(r_0 - Az_m, w_i) = 0, \quad \forall w_i \in L_m, \quad i = 1,2,\cdots,n. \tag{7.11.4}$$

下面把伽辽金原理用矩阵表示.令 $V_m=(v_1,v_2,\cdots,v_m)\in\mathbb{R}^{n\times m}$,$W_m=(w_1,w_2,\cdots,w_m)\in\mathbb{R}^{n\times m}$.由于 $z_m\in K_m$,故可把 z_m 表示成

$$z_m = V_m y_m, \tag{7.11.5}$$

其中 $y_m\in\mathbb{R}^m$.改写式(7.11.4)为

$$(W_m^{\mathrm{T}} A V_m) y_m = W_m^{\mathrm{T}} r_0. \tag{7.11.6}$$

假设 $W_m^{\mathrm{T}}AV_m$ 为非奇异矩阵,不难得到近似解 y_m,从而得到方程组(7.11.3)的近似解 z_m.理论上当且仅当 $m=n$ 时,z_m 才是精确解.

K_m 和 L_m 以及它们的基$\{v_i\}_{i=1}^m$和$\{w_i\}_{i=1}^m$的不同选取,便给出基于变分原理求解方程组(7.11.3)的不同算法.

称 $\mathrm{span}\{r_0, Ar_0, \cdots, A^{m-1}r_0\}$ 为由矩阵 A 和向量 r_0 生成的**克雷洛夫**(Krylov)**子空间**.取

$$K_m = \mathrm{span}\{r_0, Ar_0, \cdots, A^{m-1}r_0\}$$

的算法,称为**克雷洛夫子空间法**(Krylov subspace method).

在克雷洛夫子空间法中,当取 $L_m=K_m$ 时,由伽辽金原理构成**阿诺尔德**(Arnoldi)**算法**(见 7.11.3 节).当取 $L_m=AK_m$ 时,由伽辽金原理构成 **GMRES 算法**(见 7.12 节).

7.11.2 阿诺尔德过程

阿诺尔德过程是一种建立克雷洛夫子空间 K_m 的正交基的算法.它选用**格拉姆-施密特**(Gram-Schmidt)正交规范化过程,将

K_m 的基正交规范化为$\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_m$，其中$(\boldsymbol{v}_i, \boldsymbol{v}_j)=0(i\neq j)$且$\|\boldsymbol{v}_i\|_2=1(i=1,2,\cdots,m)$，称这过程为**阿诺尔德过程**. 即首先取$\boldsymbol{v}_1=\dfrac{\boldsymbol{r}_0}{\|\boldsymbol{r}_0\|_2}$，假设已有$\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_k \in \mathrm{span}\{\boldsymbol{r}_0, \boldsymbol{A}\boldsymbol{r}_0, \cdots, \boldsymbol{A}^{m-1}\boldsymbol{r}_0\}$，它们相互正交且均规范化，则$\boldsymbol{v}_{k+1}$的构造过程如下：

$$\tilde{\boldsymbol{v}}_{k+1}=\boldsymbol{A}\boldsymbol{v}_k-\sum_{i=1}^{k} h_{ik}\boldsymbol{v}_i, \tag{7.11.7}$$

其中 $h_{ik}=(\boldsymbol{A}\boldsymbol{v}_k, \boldsymbol{v}_i)$. 则取

$$\boldsymbol{v}_{k+1}=\frac{\tilde{\boldsymbol{v}}_{k+1}}{\|\tilde{\boldsymbol{v}}_{k+1}\|_2}. \tag{7.11.8}$$

令 $h_{k+1,k}=\|\tilde{\boldsymbol{v}}_{k+1}\|_2$，若 $m<n$ 且 $h_{k+1,k}\neq 0, k=1,2,\cdots,m-1$. 则完成了阿诺尔德过程.

定理 7.11.1 对于 $m<n$，阿诺尔德过程产生的向量序列$\{\boldsymbol{v}_i\}_{i=1}^{m}$是 $\mathrm{span}\{\boldsymbol{r}_0, \boldsymbol{A}\boldsymbol{r}_0, \cdots, \boldsymbol{A}^{m-1}\boldsymbol{r}_0\}$的一组标准正交基，而且$\boldsymbol{v}_{m+1}\perp \mathrm{span}\{\boldsymbol{r}_0, \boldsymbol{A}\boldsymbol{r}_0, \cdots, \boldsymbol{A}^{m-1}\boldsymbol{r}_0\}$.

过程 7.11.2 阿诺尔德过程

(1) 计算$\boldsymbol{v}_1=\dfrac{\boldsymbol{r}_0}{\|\boldsymbol{r}_0\|_2}$

(2) 对于 $j=1,2,\cdots,m$

  计算 $\boldsymbol{\omega}_j := \boldsymbol{A}\boldsymbol{v}_j$

  对于$i=1,2,\cdots,j$

    $h_{ij}=(\boldsymbol{\omega}_j, \boldsymbol{v}_i)$

    $\boldsymbol{\omega}_j := \boldsymbol{\omega}_j - h_{ij}\boldsymbol{v}_i$

    i 尾

  $h_{j+1,j}=\|\boldsymbol{\omega}_j\|_2$，若 $h_{j+1,j}=0$ 停止

  $\boldsymbol{v}_{j+1}=\boldsymbol{\omega}_j/h_{j+1,j}$

  j 尾

由式(7.11.7)和式(7.11.8)，阿诺尔德过程的矩阵表示为

$$AV_k = V_k H_k + h_{k+1,k} v_{k+1} e_k^T \tag{7.11.9}$$

其中

$$V_k = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_k \end{bmatrix}, \quad H_k = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1,k-1} & h_{1k} \\ h_{21} & h_{22} & \cdots & h_{2,k-1} & h_{2k} \\ & h_{32} & \cdots & h_{3,k-1} & h_{3k} \\ & & \ddots & \vdots & \vdots \\ & & & h_{k,k-1} & h_{kk} \end{bmatrix}.$$

7.11.3 阿诺尔德算法

在伽辽金原理中取 $K_m = L_m = \text{span}\{r_0, Ar_0, \cdots, A^{m-1}r_0\}$，其中 $r_0 = b - Ax_0$. 首先利用阿诺尔德过程 7.11.2，将 K_m 和 L_m 的基正交规范化. 其中 $v_1 = \frac{r_0}{\|r_0\|_2}$，$V_m = (v_1, v_2, \cdots, v_m)$ 为这组标准正交基构成的矩阵. 由式(7.11.9)，取 $k = m$，得

$$AV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^T. \tag{7.11.10}$$

再由伽辽金原理得到的方程组(7.11.6)，取 $W_m = V_m$，有

$$(V_m^T A V_m) y_m = V_m^T r_0,$$

令 $\beta = \|r_0\|_2$，有 $r_0 = \beta v_1$，代入上式. 又因为 $v_1 = V_m e_1$，$V_m^T v_{m+1} = 0$ 及 $V_m^T V_m = 1$，因此，上式变成

$$H_m y_m = \beta e_1, \tag{7.11.11}$$

其中 $y_m \in \mathbb{R}^m$，$H_m \in \mathbb{R}^{m \times m}$ 为一个**上海森伯格**(Hessenberg)**矩阵**. 如果 H_m 非奇异，m 比较小，则可用直接方法求解方程(7.11.11)，得 y_m，代入式(7.11.5)得 z_m，将 z_m 代入式(7.11.2)，得原方程组(7.11.1)的近似解 x_m. 称这种方法为**阿诺尔德完全正交化法**(full orthogonalization method). 简记为**阿诺尔德-FOM 算法**. 如果矩阵 H_m 奇异，则算法出现恶性中断，是这种算法很大的不足.

对于固定的 $m > 0$，阿诺尔德算法给出残余向量的大小 $\|r_0 - Az_m\|_2$.

定理 7.11.3 对于 $m>0$，有

$$\| r_0 - Az_m \|_2 = h_{m+1,m} \mid e_m^{\mathrm{T}} y_m \mid .$$

算法 7.11.4 阿诺尔德-FOM 算法

(1) 任取 $x_0 \in \mathbb{R}^n$，给定误差上界 ε

(2) 计算 $r_0 = b - Ax_0$，$\beta = \| r_0 \|_2$，$v_1 = \dfrac{r_0}{\beta}$

(3) 确定矩阵 H_m 的阶 $m \ll n$ 且置 $H_m = 0$

(4) 　　对于 $j = 1, 2, \cdots, m$

(5) 　　　　计算 $w_j := Av_j$

(6) 　　　　对于 $i = 1, 2, \cdots, j$

(7) 　　　　　　$h_{ij} = (w_j, v_j)$

(8) 　　　　　　$\omega_j := w_j - h_{ij} v_i$

(9) 　　　　　　i 尾

(10) 　　计算 $h_{j+1,j} = \| w_j \|_2$，如果 $h_{j+1,j} = 0$，则置 $m := j$ 且转向(14)

(11) 　　计算 $v_{j+1} = w_j / h_{j+1,j}$

(12) 　　j 尾

(13) 如果 H_m 奇异，则算法恶性中断，变换 x_0 转向(2)

(14) 求解 $H_m y_m = \beta e_1$，$z_m = V_m y_m$，$x_m = x_0 + z_m$. 否则

(15) 如果 $\| r_0 - Az_m \| (= h_{m+1,m} e_m^{\mathrm{T}} y_m) < \varepsilon$，则 $x = x_m$ 停止. 否则

(16) $m = m + 1$，如果 $m \leqslant n$ 则置 $H_m = 0$. 则转(4)，否则停止.

上述算法 m 愈大，格拉姆-施密特正交化愈耗时，保存 H_m 和 V_m 的存储空间也随之增加. 改进的办法是：对于大型方程组，限制 m 的最大值. 周期性地重新开始，简称**阿诺尔德-FOM(m)算法**.

算法 7.11.5 阿诺尔德-FOM(m)算法

(1) 对于固定的 $m \ll n$，任给定 $x_0 \in \mathbb{R}^n$，给定误差上界 ε.

(2) 计算 $\boldsymbol{r}_0=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_0$，$\beta=\|\boldsymbol{r}_0\|_2$，$\boldsymbol{v}_1=\dfrac{\boldsymbol{r}_0}{\beta}$.

(3) 同算法 7.11.4 的步骤(4)到(14).

(4) 如果 $\|\boldsymbol{r}_0-\boldsymbol{A}\boldsymbol{z}_m\|_2(=h_{m+1,m}\boldsymbol{e}_m^{\mathrm{T}}\boldsymbol{y}_m)<\varepsilon$，则 $\boldsymbol{x}=\boldsymbol{x}_m$ 停止.

(5) 否则 $\boldsymbol{x}_0=\boldsymbol{x}_m$ 且转向(2)

在阿诺尔德-FOM 算法的正交化过程中，$\boldsymbol{v}_i$ 和前面已经算出的所有向量 $\boldsymbol{v}_{i-1},\cdots,\boldsymbol{v}_1$ 都正交. 另一种简化的形式是在阿诺尔德过程中不要求 $\boldsymbol{v}_i$ 和 $\boldsymbol{v}_{i-1},\cdots,\boldsymbol{v}_1$ 中每个向量正交，而只要求 $\boldsymbol{v}_i$ 和 $\boldsymbol{v}_{i-1},\boldsymbol{v}_{i-2},\cdots,\boldsymbol{v}_{i-k}$ 正交. 其中 k 是一个不大的正整数. 这样得到的算法称为**阿诺尔德不完全正交**(incomplete orthogonalization)**算法**，简称**阿诺尔德-IOM 算法**. 只要选择 k，将算法 7.11.4 和算法 7.11.5 中的 $i=1,2,\cdots,j$ 改为 $i=\max\{1,j-k+1\},\cdots,j$. 并将第 10 行改为计算 $h_{j+1,j}=\|\boldsymbol{\omega}_j\|_2$，就分别得到阿诺尔德-IOM 算法和阿诺尔德-IOM(m)算法.

不完全正交化过程得到的上海森伯格矩阵 $\boldsymbol{H}_m$ 是带宽为 $k+1$ 的带结构，如 $k=3$ 和 $m=5$，

$$\boldsymbol{H}_m=\begin{pmatrix} h_{11} & h_{12} & h_{13} & & \\ h_{21} & h_{22} & h_{23} & h_{24} & \\ & h_{32} & h_{33} & h_{34} & h_{35} \\ & & h_{43} & h_{44} & h_{45} \\ & & & h_{54} & h_{55} \end{pmatrix}.$$

将其 LU 分解为 $\boldsymbol{H}_m=\boldsymbol{L}_m\boldsymbol{U}_m$，

$$\boldsymbol{H}_m=\begin{pmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ & l_{32} & 1 & & \\ & & l_{43} & 1 & \\ & & & l_{54} & 1 \end{pmatrix},\quad \boldsymbol{U}_m=\begin{pmatrix} u_{11} & u_{12} & u_{13} & & \\ & u_{22} & u_{23} & u_{24} & \\ & & u_{33} & u_{34} & u_{35} \\ & & & u_{44} & u_{45} \\ & & & & u_{55} \end{pmatrix}.$$

由式(7.11.2),式(7.11.5)和式(7.11.11)有

$$\begin{aligned} \boldsymbol{x}_m &= \boldsymbol{x}_0 + \boldsymbol{z}_m = \boldsymbol{x}_0 + \boldsymbol{V}_m \boldsymbol{y}_m \\ &= \boldsymbol{x}_0 + \boldsymbol{V}_m \boldsymbol{H}_m^{-1}(\beta \boldsymbol{e}_1) = \boldsymbol{x}_0 + \boldsymbol{V}_m \boldsymbol{U}_m^{-1} \boldsymbol{L}_m^{-1}(\beta \boldsymbol{e}_1), \\ \boldsymbol{x}_m &= \boldsymbol{x}_0 + \boldsymbol{P}_m \boldsymbol{S}_m, \end{aligned} \tag{7.11.12}$$

其中 $\boldsymbol{P}_m = \boldsymbol{V}_m \boldsymbol{U}_m^{-1}, \boldsymbol{S}_m = \boldsymbol{L}_m^{-1}(\beta \boldsymbol{e}_1)$. 因为 $\boldsymbol{U}_m$ 为带型上三角矩阵,利用 $\boldsymbol{P}_m \boldsymbol{U}_m = \boldsymbol{V}_m$,得到左式两端最后一列相等的关系式 $\sum\limits_{i=m-k+1}^{m} u_{im} \boldsymbol{p}_i = \boldsymbol{v}_m$,其等价公式为

$$\boldsymbol{p}_m = \frac{1}{u_{mm}} \left(\boldsymbol{v}_m - \sum_{i=m-k+1}^{m-1} u_{im} \boldsymbol{p}_i \right). \tag{7.11.13}$$

因为 $\boldsymbol{L}_m$ 的特殊结构,又令 $\boldsymbol{S}_m = \begin{pmatrix} \boldsymbol{S}_{m-2} \\ \rho_{m-1} \\ \rho_m \end{pmatrix}$,从 $\boldsymbol{L}_m \boldsymbol{S}_m = \beta \boldsymbol{e}_1$,得

$$\rho_m = -l_{mm-1} \rho_{m-1}. \tag{7.11.14}$$

又将 $\boldsymbol{S}_m$ 写成 $\boldsymbol{S}_m = \begin{pmatrix} \boldsymbol{S}_{m-1} \\ \rho_m \end{pmatrix}$,其中 $\boldsymbol{S}_{m-1} \in \mathbb{R}^{(m-1)\times(m-1)}$,则由式(7.11.12)得

$$\begin{aligned} \boldsymbol{x}_m &= \boldsymbol{x}_0 + (\boldsymbol{P}_{m-1}, \boldsymbol{p}_m) \begin{pmatrix} \boldsymbol{S}_{m-1} \\ \rho_m \end{pmatrix} = \boldsymbol{x}_0 + \boldsymbol{P}_{m-1} \boldsymbol{S}_{m-1} + \rho_m \boldsymbol{p}_m, \\ \boldsymbol{x}_m &= \boldsymbol{x}_{m-1} + \rho_m \boldsymbol{p}_m, \end{aligned} \tag{7.11.15}$$

其中 $\boldsymbol{x}_{m-1} = \boldsymbol{x}_0 + \boldsymbol{P}_{m-1} \boldsymbol{S}_{m-1}$. 综合以上过程得到阿诺尔德-IOM 算法.

算法 7.11.6 阿诺尔德-IOM 算法

(1) 选 $\boldsymbol{x}_0 \in \mathbb{R}^n$,给定误差上界 ε 和正整数 k.

(2) 计算 $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0, \beta = \|\boldsymbol{r}_0\|_2, \boldsymbol{v}_1 := \boldsymbol{r}_0/\beta$

(3) 确定矩阵 $\boldsymbol{H}_m$ 的阶 $m \ll n$ 且置 $\boldsymbol{H}_m = \boldsymbol{0}$

(4) 　　对于 $j = 1, 2, \cdots, m$

(5) 　　　　计算 $\boldsymbol{w}_j := \boldsymbol{A}\boldsymbol{v}_j$

(6)　　对于 $i=\max\{1,m-k+1\},\cdots,m$

(7)　　　$h_{ij}=(\boldsymbol{w}_j,\boldsymbol{v}_j)$

(8)　　　$\boldsymbol{w}_j:=\boldsymbol{w}_j-h_{ij}\boldsymbol{v}_i$

(9)　　　i 尾

(10)　　计算 $h_{j+1,j}=\|\boldsymbol{w}_j\|_2,\boldsymbol{v}_{j+1}=\boldsymbol{w}_j/h_{j+1,j}$

(11)　　j 尾

(12) 对 $\boldsymbol{H}_m$ 作 $\boldsymbol{L}_m\boldsymbol{U}_m$ 分解，如果 $u_{mm}=0$，停止；否则

(13) 计算 $\rho_m=\{$如果 $m=1$，则 β，否则 $-l_{mm-1}\rho_{m-1}\}$

(14) 计算 $p_m=u_{mm}^{-1}\left(\boldsymbol{v}_m-\sum\limits_{i=m-k+1}^{m-1}u_{im}\boldsymbol{p}_i\right)$（对于 $i\leqslant 0$，量 $u_{im}\boldsymbol{p}_i\equiv\boldsymbol{0}$）

(15) 计算 $\boldsymbol{x}_m=\boldsymbol{x}_{m-1}+\rho_m\boldsymbol{p}_m$

(16) 如果 $\|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_m\|_2<\varepsilon$，则 $\boldsymbol{x}=\boldsymbol{x}_m$ 停止，否则

(17) $m=m+1$ 置 $\boldsymbol{H}_m=\boldsymbol{0}$，如果 $m\leqslant n$ 则转(4). 否则停止

上面的算法中，用不选主元的高斯消元法对 $\boldsymbol{H}_m$ 作 LU 分解，这或许在算法的第 12 行就过早地中断. 如果改用选主元的高斯消去法，问题就解决了. 仿算法 7.11.5 修改算法 7.11.6 可得到阿诺尔德-IOM(m)算法.

阿诺尔德算法最大的不足是无法预料恶性中断何时发生. 对一般矩阵，阿诺尔德算法的收敛性至今没有给出证明. 但不少数值计算表明：循环算法即阿诺尔德-FOM(m)算法或阿诺尔德-IOM(m)算法是很有效的.

7.11.4 对称兰乔斯算法

考虑方程组(7.11.1)，其中 $\mathbf{A}\in\mathbb{R}^{n\times n}$ 为非奇异的对称矩阵. 则对称的阿诺尔德算法就简化为对称兰乔斯(Lanczos)算法(symmetric Lanczos algorithm).

定理 7.11.7　假定阿诺尔德过程应用于实对称矩阵 $\mathbf{A}$，则由过程得到的 h_{ij} 使

$$h_{ij} = 0, \quad 1 \leqslant i \leqslant j-1,$$
$$h_{j,j+1} = h_{j+1,j}, \quad j = 1,2,\cdots,m-1.$$

换句话说,阿诺尔德过程得到的矩阵 $\boldsymbol{H}_m$ 是三对角的对称矩阵.令 $\alpha_j = h_{jj}$,$\beta_j = h_{j,j+1} = h_{j+1,j}$,用 $\boldsymbol{T}_m$ 表示 $\boldsymbol{H}_m$,有

$$\boldsymbol{T}_m = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{m-1} & \alpha_{m-1} & \beta_m \\ & & & \beta_m & \alpha_m \end{pmatrix}.$$

这时,格拉姆-施密特正交化公式(7.11.7)和式(7.11.8)简化为三项递推公式

$$\beta_{j+1}\boldsymbol{v}_{j+1} = \boldsymbol{A}\boldsymbol{v}_j - \alpha_j\boldsymbol{v}_j - \beta_{j-1}\boldsymbol{v}_{j-1}. \qquad (7.11.16)$$

将阿诺尔德-FOM 算法 7.11.4 简化为如下的对称兰乔斯算法.

算法 7.11.8　对称兰乔斯算法

(1) 任给 $\boldsymbol{x}_0 \in \mathbb{R}^n$,给定误差上界 $\epsilon > 0$.

(2) 计算 $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$,$\beta = \|\boldsymbol{r}_0\|_2$,$\boldsymbol{v}_1 = \boldsymbol{r}_0/\beta$

(3) 对于 $m = 1,2,\cdots,n$

(4) 　　$\beta_1 = 0$

(5) 　　对于 $j = 1,2,\cdots,m$

(6) 　　计算 $\boldsymbol{w}_j = A\boldsymbol{v}_j - \beta_j\boldsymbol{v}_{j-1}$;$\alpha_j = (\boldsymbol{w}_j, \boldsymbol{v}_j)$,$\boldsymbol{w}_j = \boldsymbol{w}_j - \alpha_j\boldsymbol{v}_j$

(7) 　　计算 $\beta_{j+1} = \|\boldsymbol{w}_j\|_2$,如果 $\beta_{j+1} = 0$. 则 $m = j$ 且转向(11)

(8) 　　计算 $\boldsymbol{v}_{j+1} = \boldsymbol{w}_j/\beta_{j+1}$

(9) 　　j 尾

(10) 如果 $\boldsymbol{T}_m = \mathrm{tridiag}(\beta_i, \alpha_i, \beta_{i+1})$ 奇异,则更换 $\boldsymbol{x}_0$,转向(2),否则

(11) 求解三对角方程组 $\boldsymbol{T}_m\boldsymbol{y}_m = \beta\boldsymbol{e}_1$,计算 $\boldsymbol{z}_m = \boldsymbol{V}_m\boldsymbol{y}_m$

(12) 如果 $\|\boldsymbol{r}_0 - \boldsymbol{A}\boldsymbol{z}_m\|_2 < \epsilon$,则 $\boldsymbol{x} = \boldsymbol{x}_0 + \boldsymbol{z}_m$ 停止,否则

(13) m 尾

$\boldsymbol{v}_2,\cdots,\boldsymbol{v}_m)^{\mathrm{T}}$，由式(7.11.10)，即

$$\boldsymbol{A}\boldsymbol{V}_m=\boldsymbol{V}_m\boldsymbol{H}_m+h_{m+1,m}\boldsymbol{v}_{m+1}\boldsymbol{e}_m^{\mathrm{T}},$$

可改写为

$$\boldsymbol{A}\boldsymbol{V}_m=\boldsymbol{V}_{m+1}\overline{\boldsymbol{H}}_m,\tag{7.12.2}$$

其中

$$\overline{\boldsymbol{H}}_m=\begin{pmatrix}\boldsymbol{H}_m\\ h_{m+1,m}\quad\boldsymbol{e}_m^{\mathrm{T}}\end{pmatrix},\quad \boldsymbol{H}_m=\begin{pmatrix}h_{11}&h_{12}&\cdots&h_{1m}\\ h_{21}&h_{22}&\cdots&h_{2m}\\ &\ddots&\ddots&\vdots\\ &&h_{m,m-1}&h_{mm}\end{pmatrix}.\tag{7.12.3}$$

由定理 7.12.1，令 $\boldsymbol{x}=\boldsymbol{x}_0+\boldsymbol{V}_m\boldsymbol{y}$，再利用 $\boldsymbol{r}_0=\beta\boldsymbol{v}_1$，公式(7.12.2)及 $\boldsymbol{V}_{m+1}^{\mathrm{T}}\boldsymbol{V}_{m+1}=\boldsymbol{I}$，有

$$\begin{aligned}\|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}\|_2&=\|\boldsymbol{r}_0-\boldsymbol{A}\boldsymbol{V}_m\boldsymbol{y}\|_2=\|\beta\boldsymbol{v}_1-\boldsymbol{V}_{m+1}\overline{\boldsymbol{H}}_m\boldsymbol{y}\|_2\\&=\|\boldsymbol{V}_{m+1}(\beta\boldsymbol{e}_1-\overline{\boldsymbol{H}}_m\boldsymbol{y})\|_2,\end{aligned}$$

$$\|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}\|_2=\|\beta\boldsymbol{e}_1-\overline{\boldsymbol{H}}_m\boldsymbol{y}\|_2.\tag{7.12.4}$$

求解最小二乘问题 $\min\limits_{x\in x_0+K_m}\|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}\|_2$ 变成求解等价的最小二乘问题 $\min\limits_{y\in\mathbb{R}^m}\|\beta\boldsymbol{e}_1-\overline{\boldsymbol{H}}_m\boldsymbol{y}\|_2$ 并得到解 $\boldsymbol{y}_m$，表示为

$$\boldsymbol{y}_m:=\arg\min_{y\in\mathbb{R}^m}\|\beta\boldsymbol{e}_1-\overline{\boldsymbol{H}}_m\boldsymbol{y}\|_2.$$

综上所述，构成了**阿诺尔德-GMRES 算法**.

算法 7.12.2 阿诺尔德-GMRES 算法

(1) 任取 $\boldsymbol{x}_0\in\mathbb{R}^n$，给定误差上界 $\varepsilon>0$

(2) 计算 $\boldsymbol{r}_0=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_0$，$\beta=\|\boldsymbol{r}_0\|_2$，$\boldsymbol{v}_1=\dfrac{\boldsymbol{r}_0}{\beta}$

(3) 选定 $m\ll n$，$\overline{\boldsymbol{H}}_m\in\mathbb{R}^{(m+1)\times m}$，置 $\overline{\boldsymbol{H}}_m=\boldsymbol{0}$

(4) 　对于 $j=1,2,\cdots,m$

(5) 　　计算 $w_j:=\boldsymbol{A}\boldsymbol{v}_j$

(6) 　　对于 $i=1,2,\cdots,j$

7.12 广义极小残余算法

考虑大型稀疏线性方程组

$$Ax = b, \quad x, b \in \mathbb{R}^n, \tag{7.12.1}$$

其中 $A \in \mathbb{R}^{n \times n}$ 为非奇异阵且不必对称.

取 $x_0 \in \mathbb{R}^n$ 为任意向量,令

$$x = x_0 + z,$$

则有等价的方程组

$$Az = r_0,$$

其中 $r_0 = b - Ax_0$.

取 $K_m = \text{span}\{r_0, Ar_0, \cdots, A^{m-1}r_0\}$ 和 $L_m = \text{span}\{Ar_0, A^2r_0, \cdots, A^m r_0,\}$,简记 $L_m = AK_m$. 仍利用伽辽金原理(见 7.11.1 节),取 $z_m \in K_m$,使 $r_m = r_0 - Az_m$ 与 L_m 中所有的向量正交而求得 z_m,这就构成了**广义极小残余算法**(generalized minimal residual algorithm),简称 **GMRES 算法**. 有如下定理.

定理 7.12.1 令 A 为 $n \times n$ 方阵并假定 $L_m = AK_m$,取任意 $x_0 \in \mathbb{R}^n$ 为初始向量,则按伽辽金原理计算近似解 $\tilde{x}$,等价于 $\tilde{x}$ 在 $x_0 + K_m$ 中极小化 $\| b - Ax \|_2$,即

$$\| b - A\tilde{x} \|_2 = \min_{x \in x_0 + k_m} \| b - Ax \|_2.$$

下面分别用两种方法完成 GMRES 算法中的阿诺尔德过程. 一种是用格拉姆-施密特正交化方法,构成阿诺尔德-GMRES 算法;另一种是用豪斯霍尔德正交化方法,构成豪斯霍尔德-GMRES 算法.

7.12.1 阿诺尔德-GMRES 算法

首先用阿诺尔德过程 7.11.2,得到 K_m 的一组标准正交基 $\{v_i\}_{i=1}^m$. 其中 $v_1 = \frac{r_0}{\| r_0 \|_2}$, $\beta = \| r_0 \|_2$,有 $r_0 = \beta v_1$,令 $V_m = (v_1,$,

(7) $h_{ij}=(\boldsymbol{w}_j,\boldsymbol{v}_i)$

(8) $\boldsymbol{w}_j:=\boldsymbol{w}_j-h_{ij}\boldsymbol{v}_i$

(9) i 尾

(10) 计算 $h_{j+1,j}=\|\boldsymbol{w}_j\|_2$，如果 $h_{j+1,j}=0$，则 $m:=j$ 且转向(13)，否则

(11) 计算 $\boldsymbol{v}_{j+1}=\boldsymbol{w}_j/h_{j+1,j}$

(12) j 尾

(13) 计算 $\boldsymbol{y}_m:=\arg\min\limits_{y\in\mathbb{R}^m}\|\beta\boldsymbol{e}_1-\overline{\boldsymbol{H}}_m\boldsymbol{y}\|_2$，$\boldsymbol{z}_m=\boldsymbol{V}_m\boldsymbol{y}_m$，$\boldsymbol{x}_m=\boldsymbol{x}_0+\boldsymbol{z}_m$

(14) 如果 $\|\boldsymbol{r}_0-\boldsymbol{A}\boldsymbol{z}_m\|_2<\varepsilon$，则 $\boldsymbol{x}=\boldsymbol{x}_m$，停止，否则

(15) $m=m+1$，如果 $m\leqslant n$ 则置 $\overline{\boldsymbol{H}}_m=\boldsymbol{0}$，转向(4)，否则停止

上述算法当 m 很大时，阿诺尔德过程耗时且 $\boldsymbol{V}_m$ 和 $\overline{\boldsymbol{H}}_m$ 的存储量很大．所以对于大型方程组，限制 m 的最大值．取固定的 $m<n$，周期性地重新开始，简称这种算法为**阿诺尔德-GMRES(m)算法**．

算法 7.12.3 阿诺尔德-GMRES(m)算法

(1) 对于固定的 $m<n$，任给 $\boldsymbol{x}_0\in\mathbb{R}^n$，给定误差上界 $\varepsilon>0$

(2) 计算 $\boldsymbol{r}_0=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_0$，$\beta=\|\boldsymbol{r}_0\|_2$，$\boldsymbol{v}_1=\dfrac{\boldsymbol{r}_0}{\beta}$

(3) 同算法 7.12.2 的步骤(4)到步骤(14)

(4) $\boldsymbol{x}_0=\boldsymbol{x}_m$ 且转向(2)

7.12.2 豪斯霍尔德-GMRES 算法

下面用豪斯霍尔德正交变换，找克雷洛夫子空间 $K_m=\text{span}\{\boldsymbol{r}_0,\boldsymbol{A}\boldsymbol{r}_0,\cdots,\boldsymbol{A}^{m-1}\boldsymbol{r}_0\}$的一组标准正交基$\{\boldsymbol{v}_i\}_{i=1}^m$，其中 $\boldsymbol{r}_0=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_0$．豪斯霍尔德正交化是用反射矩阵 $\boldsymbol{P}_j$ 将矩阵 $\boldsymbol{X}$ 变换为上三角矩阵．

$$\boldsymbol{P}_j=\boldsymbol{I}-2\boldsymbol{w}_j\boldsymbol{w}_j^{\mathrm{T}}$$

其中 $w_j = \frac{z}{\|z\|_2}, \quad z = (z_1, z_2, \cdots, z_n)^T.$ (7.12.5)

$$z_i = \begin{cases} 0, & i < j, \\ \beta + x_{ii}, & i = j, \quad \beta = \text{sign}(x_{jj})\left(\sum_{i=j}^{n} x_{ij}^2\right)^{\frac{1}{2}}. \\ x_{ij}, & i > j, \end{cases}$$

下面用豪斯霍尔德正交化方法完成阿诺尔德过程，得到与阿诺尔德-FOM 算法 7.11.4 对应的豪斯霍尔德-GMRES 算法，或者说 m 从 1 到 n 的完全豪斯霍尔德-GMRES 算法. 也有与阿诺尔德-FOM(m)算法 7.11.5 相对应的豪斯霍尔德-GMRES(m)算法，或者说固定 m，重新开始的算法，仅列出后者.

算法 7.12.4　豪斯霍尔德-GMRES(m)算法

(1) 任选 $x_0 \in \mathbb{R}^n$. 给定误差上界 ϵ，选定 m.

(2) 计算 $r_0 = b - Ax_0, z = r_0$

(3) 对于 $j = 1, 2, \cdots, m, m+1$

(4) 　　计算豪斯霍尔德向量 w_j（用式(7.12.5)），使

(5) 　　$(w_j)_i = 0, i = 1, 2, \cdots, j-1$ 且

(6) 　　$(P_j z)_i = 0, i = j+1, \cdots, n$，其中 $P_j = I - 2w_j w_j^T$

(7) 　　$h_{j-1} = P_j z$，如果 $j = 1$，则令 $\beta = e_1^T h_0$

(8) 　　$v = P_1 P_2 \cdots P_j e_j$

(9) 　　如果 $j \leqslant m$ 计算 $z = P_j P_{j-1} \cdots P_1 A v$

(10) 　　j 尾

(11) 定义 $\overline{H}_m = (h_1, \cdots, h_m)$ 是 $(m+1) \times m$ 矩阵

(12) 计算 $y_m = \arg \min_{y \in \mathbb{R}^m} \| \beta e_1 - \overline{H}_m y \|_2, y_m = (\eta_1, \eta_2, \cdots, \eta_m)^T$

(13) $z = 0$

(14) 对于 $j = m, m-1, \cdots, 1$

(15) 　　$z = P_j(\eta_j e_j + z)$

(16) 　　j 尾

(17) 计算 $x_m = x_0 + z$

(18) 如果 $\| \boldsymbol{r}_0 - \boldsymbol{A}\boldsymbol{z} \|_2 < \varepsilon$,则 $\boldsymbol{x} = \boldsymbol{x}_m$ 停止

(19) $\boldsymbol{x}_0 = \boldsymbol{x}_m$ 且转向(2)

该算法中(2)～(10)是用一系列的豪斯霍尔德正交变换 $\boldsymbol{P}_j(j=1,2,\cdots,m)$完成了对矩阵 $\boldsymbol{X}=(\boldsymbol{r}_0,\boldsymbol{A}\boldsymbol{v}_1,\boldsymbol{A}\boldsymbol{v}_2,\cdots,\boldsymbol{A}\boldsymbol{v}_m)$的 QR 因式分解,即

$$\boldsymbol{Q}_m(\boldsymbol{r}_0,\boldsymbol{A}\boldsymbol{v}_1,\cdots,\boldsymbol{A}\boldsymbol{v}_m) = (\boldsymbol{h}_0,\boldsymbol{h}_1,\cdots,\boldsymbol{h}_m), \quad (7.12.6)$$

其中 $\boldsymbol{Q}_m = \boldsymbol{P}_m\boldsymbol{P}_{m-1}\cdots\boldsymbol{P}_1$ 为酉阵(即 $\boldsymbol{Q}_m^{\mathrm{H}}\boldsymbol{Q}_m = \boldsymbol{I}$,$\boldsymbol{Q}_m^{\mathrm{H}}$ 为 $\boldsymbol{Q}_m$ 的共轭转置),$(\boldsymbol{h}_0,\boldsymbol{h}_1,\cdots,\boldsymbol{h}_m)$是 $n\times(m+1)$的上三角矩阵.由算法 7.12.4 的(6)～(8),有

$$\boldsymbol{h}_j = \boldsymbol{P}_{j+1}\boldsymbol{z} = \boldsymbol{P}_{j+1}\boldsymbol{Q}_j\boldsymbol{A}\boldsymbol{v}_j = \boldsymbol{Q}_{j+1}\boldsymbol{A}\boldsymbol{v}_j. \quad (7.12.7)$$

注意到向量 $\boldsymbol{h}_j$ 的第 $j+2$ 到 n 的分量为零,则对 $i \geqslant j+2$,有 $\boldsymbol{P}_i\boldsymbol{h}_j = \boldsymbol{h}_j$,因此

$$\boldsymbol{h}_j = \boldsymbol{P}_m\boldsymbol{P}_{m-1}\cdots\boldsymbol{P}_{j+2}\boldsymbol{h}_j = \boldsymbol{Q}_m\boldsymbol{A}\boldsymbol{v}_j, \quad j = 1,2,\cdots,m,$$

这就得到式(7.12.6).

由式(7.12.7)得

$$\begin{aligned}\boldsymbol{A}\boldsymbol{v}_j &= \boldsymbol{Q}_{j+1}^{\mathrm{T}}\sum_{i=1}^{j+1}h_{ij}\boldsymbol{e}_i = \sum_{i=1}^{j+1}h_{ij}\boldsymbol{Q}_{j+1}^{\mathrm{T}}\boldsymbol{e}_i \\ &= \sum_{i=1}^{j+1}h_{ij}\,\boldsymbol{v}_i, \quad j = 1,2,\cdots,m,\end{aligned}$$

显然有

$$\boldsymbol{A}\boldsymbol{V}_m = \boldsymbol{V}_{m+1}\overline{\boldsymbol{H}}_m.$$

它与 7.12.1 节的公式(7.12.2)相同,其中 $\overline{\boldsymbol{H}}_m$ 见式(7.12.3).同理可得到与式(7.12.4)相同的公式

$$\| \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x} \|_2 = \| \beta\boldsymbol{e}_1 - \overline{\boldsymbol{H}}_m\boldsymbol{y} \|_2.$$

求解最小二乘问题$\min\limits_{y\in \mathrm{R}^m} \| \beta\boldsymbol{e}_1 - \overline{\boldsymbol{H}}_m\boldsymbol{y} \|_2$,得到解

$$\boldsymbol{y}_m := \arg\min_{y\in \mathrm{R}^m} \| \beta\boldsymbol{e}_1 - \overline{\boldsymbol{H}}_m\boldsymbol{y} \|_2,$$

其中 $\boldsymbol{y}_m = (\eta_1,\eta_2,\cdots,\eta_m)^{\mathrm{T}}$.从而有

$$\boldsymbol{x}_m = \boldsymbol{x}_0 + \boldsymbol{V}_m \boldsymbol{y}_m, \tag{7.12.8}$$

其中 $\boldsymbol{V}_m=(\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_m)$，$\boldsymbol{v}_j=\boldsymbol{P}_1\boldsymbol{P}_2\cdots\boldsymbol{P}_j\boldsymbol{e}_j$. 上式的另一种计算形式

$$\begin{aligned}\boldsymbol{x}_m &= \boldsymbol{x}_0 + \eta_1 \boldsymbol{v}_1 + \cdots + \eta_m \boldsymbol{v}_m \\ &= \boldsymbol{x}_0 + \eta_1 \boldsymbol{P}_1 \boldsymbol{e}_1 + \cdots + \eta_m \boldsymbol{P}_1 \boldsymbol{P}_2 \cdots \boldsymbol{P}_m \boldsymbol{e}_m \\ &= \boldsymbol{x}_0 + \boldsymbol{P}_1(\eta_1 \boldsymbol{e}_1 + \boldsymbol{P}_2(\eta_2 \boldsymbol{e}_2 + \cdots + \\ &\quad \boldsymbol{P}_{m-1}(\eta_{m-1}\boldsymbol{e}_{m-1} + \boldsymbol{P}_m \eta_m \boldsymbol{e}_m) + \cdots)).\end{aligned} \tag{7.12.9}$$

算法 7.12.4 的(13)～(16)行按式(7.12.9)计算 $\boldsymbol{x}_m$.

7.12.3 $\|\beta\boldsymbol{e}_1-\overline{\boldsymbol{H}}_m\boldsymbol{y}\|_2$ 极小化算法

在 GMRES 算法 7.12.2 到算法 7.12.4 中，如何有效地求解 $\|\beta\boldsymbol{e}_1-\overline{\boldsymbol{H}}_m\boldsymbol{y}\|_2$ 的极小化问题是算法是否成功的关键. 即用最小二乘法求解超定方程组

$$\overline{\boldsymbol{H}}_m \boldsymbol{y} = \beta \boldsymbol{e}_1, \tag{7.12.10}$$

其中 $\overline{\boldsymbol{H}}_m$ 是$(m+1)\times m$ 的上海森伯格矩阵，考虑到 $\overline{\boldsymbol{H}}_m$ 的特殊形式，有效的方法是用平面旋转矩阵对 $\overline{\boldsymbol{H}}_m$ 作 QR 分解，将 $\overline{\boldsymbol{H}}_m$ 变换成上三角矩阵.

定义 7.12.5 设

$$\boldsymbol{\Omega}_i = \begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & & c_i & s_i & & \\ & & & -s_i & c_i & & \\ & & & & & 1 & \\ & & & & & & \ddots \\ & & & & & & & 1 \end{pmatrix} \begin{matrix} \\ \\ \\ i\text{ 行} \\ i+1\text{ 行} \\ \\ \\ \\ \end{matrix}$$

且有 $c_i^2+s_i^2=1$，称 $\boldsymbol{\Omega}_i$ 为**平面旋转矩阵**，也称为**吉文斯**(Givens)**变换矩阵**.

用一系列 $\boldsymbol{\Omega}_i\in\mathbb{R}^{(m+1)\times(m+1)}$ $(i=1,2,\cdots,m)$，分别左乘 $\overline{\boldsymbol{H}}_m$ 和 $\beta\boldsymbol{e}_1$. 每次消去 $h_{i+1,i}$ $(i=1,2,\cdots,m)$，则有

$$s_i = \frac{h_{i+1,i}}{\sqrt{(h_{ii}^{(i-1)})^2 + h_{i+1,i}^2}}, \quad c_i = \frac{h_{ii}^{(i-1)}}{\sqrt{(h_{ii}^{(i-1)})^2 + h_{i+1,i}^2}}, \tag{7.12.11}$$

其中 $h_{ii}^{(i-1)}$ 是 $\overline{\boldsymbol{H}}_m$ 经 $i-1$ 次平面旋转后的第 $i-1$ 个对角线元素. 经 m 次旋转后,

$$\boldsymbol{Q}_m = \boldsymbol{\Omega}_m \boldsymbol{\Omega}_{m-1} \cdots \boldsymbol{\Omega}_1,$$
$$\overline{\boldsymbol{R}}_m = \boldsymbol{Q}_m \overline{\boldsymbol{H}}_m,$$
$$\bar{\boldsymbol{g}}_m = \boldsymbol{Q}_m(\beta \boldsymbol{e}_1) = (\nu_1, \cdots, \nu_{m+1})^{\mathrm{T}}$$

因为 $\boldsymbol{Q}_m$ 是酉矩阵,即 $\boldsymbol{Q}_m^{\mathrm{T}}\boldsymbol{Q}_m = \boldsymbol{I}$,所以

$$\begin{aligned} \min \| \beta \boldsymbol{e}_1 - \overline{\boldsymbol{H}}_m \boldsymbol{y} \|_2 &= \min \| \boldsymbol{Q}_m (\beta \boldsymbol{e}_1 - \overline{\boldsymbol{H}}_m \boldsymbol{y}) \|_2 \\ &= \min \| \bar{\boldsymbol{g}}_m - \overline{\boldsymbol{R}}_m \boldsymbol{y} \|_2. \end{aligned} \tag{7.12.12}$$

上式右端的极小化问题非常容易求解. 因为

$$\bar{\boldsymbol{g}}_m - \overline{\boldsymbol{R}}_m \boldsymbol{g} = \begin{pmatrix} \nu_1 \\ \nu_2 \\ \vdots \\ \nu_{m+1} \end{pmatrix} - \begin{pmatrix} \boldsymbol{R}_m \\ \boldsymbol{0} \end{pmatrix} \boldsymbol{y} = \boldsymbol{0}, \tag{7.12.13}$$

其中 $\boldsymbol{R}_m$ 是非奇异的上三角方阵. 则有

$$\boldsymbol{R}_m \boldsymbol{y} = \begin{pmatrix} \nu_1 \\ \nu_2 \\ \vdots \\ \nu_m \end{pmatrix}, \tag{7.12.14}$$

这样用最小二乘法求解超定方程组(7.12.10)的问题转为求解超定方程组(7.12.13)问题,又变化为用直接法求解三角形方程组(7.12.14)的问题. 易得方程(7.12.14)的解为 $\boldsymbol{y}_m$. 而且最小二乘残余为

$$\| \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_m \|_2 = | \nu_{m+1} |. \tag{7.12.15}$$

因此,取定 m 后,判断 GMRES 算法 7.12.2 到算法 7.12.4 终止的标准,无需计算 $\| \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_m \|_2 = \| \boldsymbol{r}_0 - \boldsymbol{A}\boldsymbol{z}_m \|_2$,只需判断 $|\nu_{m+1}| < \varepsilon$ 是否

满足即可.

7.12.4 GMRES 算法的收敛性和中断

将阿诺尔德-GMRES(m)算法和豪斯霍尔德-GMRES(m)算法合称为 GMRES(m)算法.

定义 7.12.6 设 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$(不必是对称的),如果 $\boldsymbol{A}+\boldsymbol{A}^{\mathrm{T}}$ 是一个对称正定矩阵,则称 $\boldsymbol{A}$ 是**正定矩阵**.

定理 7.12.7 设 $\boldsymbol{Ax}=\boldsymbol{b}$ 中的 $\boldsymbol{A}$ 是正定的,则对任何 $m\geqslant 1$ 的整数和任意初值 $\boldsymbol{x}_0$,GMRES(m)算法是收敛的.

定理 7.12.8 如果 $\boldsymbol{Ax}=\boldsymbol{b}$ 中的 $\boldsymbol{A}$ 是对称正定的,则对任何 $m\geqslant 1$的整数和任意初值 $\boldsymbol{x}_0$,GMRES(m)算法是收敛的.

定理 7.12.9 考虑 $\boldsymbol{Ax}=\boldsymbol{b}$,假设

(1) $\boldsymbol{A}$ 是可对角化的,即存在非奇异矩阵 $\boldsymbol{X}$,使得 $\boldsymbol{A}=\boldsymbol{X\Lambda X}^{-1}$ 成立,其中 $\boldsymbol{\Lambda}=\mathrm{diag}(\lambda_1,\lambda_2,\cdots,\lambda_n)$,$\lambda_i(i=1,\cdots,n)$为 $\boldsymbol{A}$ 的特征值.

(2) $\boldsymbol{A}$ 的全部特征值落在椭圆 $E(c,a,d)$内部. c,a,d 分别代表椭圆中心、焦距和长轴,而且 $0\notin E(c,a,d)$,则

$$\|\boldsymbol{b}-\boldsymbol{Ax}_m\|_2\leqslant \mathrm{cond}_2(\boldsymbol{X})\left|\frac{C_m\left(\dfrac{a}{d}\right)}{C_m\left(\dfrac{c}{d}\right)}\right|\|\boldsymbol{b}-\boldsymbol{Ax}_0\|_2. \tag{7.12.16}$$

该定理中 $C_m(\cdot)$是 m 阶切比雪夫多项式,有

$$\left|\frac{C_m\left(\dfrac{a}{d}\right)}{C_m\left(\dfrac{c}{d}\right)}\right|\approx\left[\frac{a+\sqrt{a^2-d^2}}{c+\sqrt{c^2-d^2}}\right]^m. \tag{7.12.17}$$

因为 $c>a$,故上式右端因子小于 1. 显然,当 m 足够大时,可直接从式(7.12.16)推出,GMRES(m)算法对任意初值 $\boldsymbol{x}_0$ 是收敛的.

在更弱的条件下,GMRES(m)算法有如下收敛性.

定理 7.12.10 考虑 $\boldsymbol{Ax}=\boldsymbol{b}$,假设 $\boldsymbol{A}$ 是可对角化的,即存在非奇异矩阵 $\boldsymbol{X}$,使得 $\boldsymbol{A}=\boldsymbol{X\Lambda X}^{-1}$,其中 $\boldsymbol{\Lambda}=\mathrm{diag}(\lambda_1,\lambda_2,\cdots,\lambda_n)$,$\lambda_i(i=1,2,\cdots,n)$为 $\boldsymbol{A}$ 的特征值,则对充分大的 m,GMRES(m)收敛.

若 n 阶矩阵 $\boldsymbol{A}$ 满足 $\boldsymbol{AA}^{\mathrm{H}}=\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}$,称 $\boldsymbol{A}$ 为正规矩阵.对角矩阵、实对称矩阵($\boldsymbol{A}^{\mathrm{T}}=\boldsymbol{A}$)、埃尔米特矩阵($\boldsymbol{A}^{\mathrm{H}}=\boldsymbol{A}$)、正交矩阵($\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}=\boldsymbol{I}$)和酉矩阵($\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}=\boldsymbol{I}$)都是正规矩阵.

$\boldsymbol{A}$ 为正规矩阵的充分必要条件是 $\boldsymbol{A}$ 酉相似于一个对角矩阵 $\boldsymbol{\Lambda}$,即存在酉矩阵 $\boldsymbol{U}$,使

$$\boldsymbol{A}=\boldsymbol{U\Lambda U}^{\mathrm{H}}$$

其中 $\boldsymbol{\Lambda}=\mathrm{diag}(\lambda_1,\lambda_2,\cdots,\lambda_n)$,$\lambda_i(i=1,\cdots,n)$为 $\boldsymbol{A}$ 的特征值.

所以正规矩阵是可对角化的.满足定理 7.12.10 的条件,对充分大的 m,求解 $\boldsymbol{Ax}=\boldsymbol{b}$ 的 GMRES(m)是收敛的.

阿诺尔德-FOM 算法(见 7.11 节)会产生恶性中断,而 GMRES 算法克服了这个困难.豪斯霍尔德-GMRES 算法不会产生中断.阿诺尔德-GMRES 算法的中断只可能在阿诺尔德过程中的第 j 步 $h_{j+1,j}=0$ 时出现中断,这时不可能求得下一步的阿诺尔德向量了,这种情况下求得的解是精确的.有以下命题.

命题 7.12.11 令 $\boldsymbol{A}$ 为非奇异阵,则求解 $\boldsymbol{Ax}=\boldsymbol{b}$ 的阿诺尔德-GMRES 算法在第 j 步中断,即 $h_{j+1,j}=0$ 当且仅当近似解 $\boldsymbol{x}_j$ 是精确的:

$$\boldsymbol{b}-\boldsymbol{Ax}_j=\boldsymbol{0}\Leftrightarrow h_{j+1,j}=0.$$

7.12.5 分裂预处理 GMRES 算法

预处理是使 GMRES 算法有效的另一个关键因素.对于 GMRES 或其他非对称迭代法,它们的预处理方法类似于预处理共轭梯度法(见 7.10 节).7.10 节的预处理方法都可以用于 GMRES 算法上.这里只给出一种类似于 ICCG 法的分裂预处理技巧,从而有分裂预处理 GMRES 算法(split preconditioning GMRES algorithm).

我们得到一非奇异矩阵 $\boldsymbol{M}$，它在某种意义下逼近 $\boldsymbol{Ax}=\boldsymbol{b}$ 中的矩阵 $\boldsymbol{A}$. 比如，对 $\boldsymbol{A}$ 作不完全 LU 分解(见 7.6.2 节)，有

$$\boldsymbol{A} = \boldsymbol{LU} - \boldsymbol{R}.$$

取预处理器 $\boldsymbol{M}=\boldsymbol{LU}\approx\boldsymbol{A}$. 与 $\boldsymbol{Ax}=\boldsymbol{b}$ 等价的方程组

$$\boldsymbol{Fy} = \boldsymbol{g}, \tag{7.12.18}$$

其中 $\boldsymbol{F}=\boldsymbol{L}^{-1}\boldsymbol{AU}^{-1}$，$\boldsymbol{y}=\boldsymbol{Ux}$，$\boldsymbol{g}=\boldsymbol{L}^{-1}\boldsymbol{b}$.

当 $\boldsymbol{LU}$ 很近似于 $\boldsymbol{A}$ 的完全分解($\boldsymbol{R}=\boldsymbol{0}$)时，$\boldsymbol{F}$ 近似于单位矩阵 $\boldsymbol{I}$. 对方程组(7.12.18)写出 GMRES 迭代算法，并整理，使得只存储原始变量.

算法 7.12.12 分裂预处理阿诺尔德-GMRES(m)算法

(1) 选 $\boldsymbol{x}_0\in\mathbb{R}^n$ 和适当大小的 $m<n$，给出误差上界 $\varepsilon>0$

(2) 计算 $\boldsymbol{r}_0=\boldsymbol{b}-\boldsymbol{Ax}_0$，$\hat{\boldsymbol{r}}_0=\boldsymbol{L}^{-1}\boldsymbol{r}_0$，$\beta=\|\hat{\boldsymbol{r}}_0\|_2$，$\boldsymbol{v}_1=\dfrac{\hat{\boldsymbol{r}}_0}{\beta}$

(3) 对于 $j=1,2,\cdots,m$

(4) 　　计算 $\boldsymbol{w}_j:=\boldsymbol{L}^{-1}\boldsymbol{AU}^{-1}\boldsymbol{v}_j$

(5) 　　对于 $i=1,\cdots,j$

(6) 　　　　$h_{ij}:=(\boldsymbol{w}_j,\boldsymbol{v}_j)$

(7) 　　　　$\boldsymbol{w}_j:=\boldsymbol{w}_j-h_{ij}\boldsymbol{v}_i$

(8) 　　　　i 尾

(9) 　　$h_{j+1,j}=\|\boldsymbol{w}_j\|_2$，如果 $h_{j+1,j}=0$，则 $m=j$ 且转向(12)

(10) 　　$\boldsymbol{v}_{j+1}=\boldsymbol{w}_j/h_{j+1,j}$

(11) 　　j 尾

(12) 计算 $\boldsymbol{y}_m=\arg\min\limits_{y\in\mathbb{R}^m}\|\beta\boldsymbol{e}_1-\overline{\boldsymbol{H}}_m\boldsymbol{y}\|_2$

(13) $\boldsymbol{x}_m=\boldsymbol{x}_0+\boldsymbol{U}^{-1}\boldsymbol{V}_m\boldsymbol{g}_m$

(14) 如果 $\|\boldsymbol{L}^{-1}(\boldsymbol{b}-\boldsymbol{Ax}_m)\|_2<\varepsilon$，则 $\boldsymbol{x}=\boldsymbol{x}_m$ 停止，否则

(15) $\boldsymbol{x}_0=\boldsymbol{x}_m$ 且转向(2).

8 矩阵特征值问题

本章讨论标准(standard)**特征值问题** $Ax=\lambda x$ 的数值方法并简单介绍广义(gerneralized)**特征值问题** $Ax=\lambda Bx$ 或 $ABx=\lambda x$ 的有效解法. 在8.1节和8.2节中,给出必要的数学基础知识,以便导出和分析后面的求特征值的方法.

8.1 基本性质

8.1.1 特征值的性质

矩阵 $A\in\mathbb{C}^{n\times n}$ 的**特征值**是其特征多项式 $P(\lambda)=\mathrm{def}(\lambda I-A)$ 的 n 个根,这些根的集合称为**谱**,记为 $\sigma(A)$. 如果 $\lambda\in\sigma(A)$,则满足

$$Ax=\lambda x \tag{8.1.1}$$

的非零向量 $x\in\mathbb{C}^n$ 称为 A 的**特征向量**,也称 x 为 A 的**右特征向量**;满足

$$y^{\mathrm{H}}A=\lambda y^{\mathrm{H}} \tag{8.1.2}$$

的非零向量 $y\in\mathbb{C}^n$ 称为 A 的**左特征向量**. 除非特别声明,特征向量一般指右特征向量.

特征多项式根 λ 的重数称为特征值 λ 的**代数重数**(algebraic multiplicity),而 λ 所对应的线性无关的特征向量的个数称为**特征值 λ 的几何重数**(geometric multiplicity). 显然几何重数不会超过代数重数. 若几何重数小于代数重数,则称 A 是(对该特征值)**亏损的**(defective).

8.1.2 特征值的界限

在某些情况下，不需要得到高精度的特征值，只需要知道一些相当粗的信息：特征值的上界或在复平面上的局部范围.

任一 $n\times n$ 矩阵 $\boldsymbol{A}$ 的任一矩阵范数 $\|\boldsymbol{A}\|$ 不小于 $\boldsymbol{A}$ 的谱半径 $\rho(\boldsymbol{A})$，即

$$\rho(\boldsymbol{A})\leqslant\|\boldsymbol{A}\|,$$

其中 $\rho(\boldsymbol{A})=\max\limits_{1\leqslant i\leqslant n}|\lambda_i|$，$\lambda_i(i=1,2,\cdots,n)$ 为 $\boldsymbol{A}$ 的特征值. 所以，$\boldsymbol{A}$ 的特征值均位于以原点为中心，以 $\|\boldsymbol{A}\|$ 为半径的复平面的一个圆盘上.

定理 8.1.1（格尔施戈林（Gerschgorin）定理） 矩阵 $\boldsymbol{A}=(a_{ij})_n$ 的任一特征值至少位于复平面 n 个圆盘

$$D_i:\left\{z\,\middle|\,|z-a_{ii}|\leqslant\sum_{\substack{j=1\\j\neq i}}^{n}|a_{ij}|\right\},i=1,2,\cdots,n$$

中的一个圆盘上.

定理 8.1.2 如果 $\boldsymbol{A}$ 的 n 个圆盘中的 m 个圆盘形成一个连通区域，且与其余的 $n-m$ 个圆盘不相连接，则在这个连通区域中恰好有 $\boldsymbol{A}$ 的 m 个特征值.

当 $m=1$ 时，即每个孤立圆盘中恰好有一个特征值.

8.1.3 极值定理

当 $\boldsymbol{A}$ 是对称阵，或更一般地说，$\boldsymbol{A}$ 是埃尔米特阵时，成立下列极小-极大定理.

定理 8.1.3 设 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$，且 $\boldsymbol{A}$ 为对称阵，则 $\boldsymbol{A}$ 的特征值是实的，可排列为 $\lambda_1\geqslant\lambda_2\geqslant\cdots\geqslant\lambda_n$，对应的特征向量构成一个正交向量组 $\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n\}$，且存在正交阵 $\boldsymbol{U}$，使 $\boldsymbol{U}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{U}$ 为对角阵，并有

$$\lambda_i=\max_{\dim(W)=i}\min_{\substack{x\in W\\x\neq 0}}\frac{(\boldsymbol{A}\boldsymbol{x},\boldsymbol{x})}{(\boldsymbol{x},\boldsymbol{x})}=\min_{x\in\operatorname{span}\{x_1,x_2,\cdots,x_i\}}\frac{(\boldsymbol{A}\boldsymbol{x},\boldsymbol{x})}{(\boldsymbol{x},\boldsymbol{x})},\tag{8.1.3}$$

$$\lambda_i = \min_{\dim(W)=n-i+1} \max_{\substack{x \in W \\ x \neq 0}} \frac{(\boldsymbol{Ax},\boldsymbol{x})}{(\boldsymbol{x},\boldsymbol{x})} = \max_{x \in \operatorname{span}\{x_i,\cdots,x_n\}} \frac{(\boldsymbol{Ax},\boldsymbol{x})}{(\boldsymbol{x},\boldsymbol{x})}, \tag{8.1.4}$$

$$\lambda_n \leqslant \frac{(\boldsymbol{Ax},\boldsymbol{x})}{(\boldsymbol{x},\boldsymbol{x})} \leqslant \lambda_1, \forall \boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{x} \neq \boldsymbol{0}. \tag{8.1.5}$$

其中式(8.1.4)当 $i=1$ 时有

$$\lambda_1 = \max_{\substack{x \in \mathbb{R}^n \\ x \neq 0}} \frac{(\boldsymbol{Ax},\boldsymbol{x})}{(\boldsymbol{x},\boldsymbol{x})} = \frac{(\boldsymbol{Ax}_1,\boldsymbol{x}_1)}{(\boldsymbol{x}_1,\boldsymbol{x}_1)}, \tag{8.1.6}$$

其中 $\boldsymbol{x}_1$ 为对应于 λ_1 的特征向量,式(8.1.3)当 $i=n$ 时,有

$$\lambda_n = \min_{\substack{x \in \mathbb{R}^n \\ x \neq 0}} \frac{(\boldsymbol{Ax},\boldsymbol{x})}{(\boldsymbol{x},\boldsymbol{x})} = \frac{(\boldsymbol{Ax}_n,\boldsymbol{x}_n)}{(\boldsymbol{x}_n,\boldsymbol{x}_n)}. \tag{8.1.7}$$

称比值$\frac{(\boldsymbol{Ax},\boldsymbol{x})}{(\boldsymbol{x},\boldsymbol{x})}$为瑞利商.

定理 8.1.4(柯西交错定理) 设 $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ 是埃尔米特阵,其特征值排列为 $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n$. 若 $\boldsymbol{A}_m \in \mathbb{C}^{m\times m}$ 是 $\boldsymbol{A}$ 的 $m(m=1,2,\cdots,n)$ 阶主子阵,其特征值排列为 $\mu_1 \geqslant \mu_2 \geqslant \cdots \geqslant \mu_m$,则

$$\mu_j \leqslant \lambda_j, \mu_{m-j+1} \geqslant \lambda_{n-j+1}, \quad j = 1,2,\cdots,m.$$

8.1.4 扰动和敏感性

实际求解特征值问题(8.1.1)时,若矩阵 $\boldsymbol{A}$ 的一个小扰动 $\delta\boldsymbol{A}$ 引起特征值(特征向量)的大变化,则称该特征值问题是**敏感的**(sensitive),或**病态的**(ill-conditioned),反之为**良态的**(well-conditioned).

例 8.1.5 设 $\boldsymbol{A} = \begin{pmatrix} 0 & 10^8 \\ 0 & 0 \end{pmatrix}$ 和扰动 $\delta\boldsymbol{A} = \begin{pmatrix} 0 & 0 \\ 10^{-10} & 0 \end{pmatrix}$. $\boldsymbol{A}$ 的特征值 $\lambda_1 = \lambda_2 = 0$. $\boldsymbol{A}+\delta\boldsymbol{A}$ 的特征值 $\mu_1 = \mu_2 = 10^{-1}$. 可见矩阵 $\boldsymbol{A}$ 的元素有 10^{-10} 的微小变化,导致其特征值的大变化 10^{-1}. 所以 $\boldsymbol{A}$ 对计算其特征值问题是病态的. 另一方面,若扰动处于 $\boldsymbol{A}$ 的严格上三

角部分，则 $\boldsymbol{A}$ 的特征值不变. 若扰动处在 $\boldsymbol{A}$ 的对角线上，则特征值的变化与扰动一样. 所以特征值问题的扰动性态很复杂. 例 8.1.5 中的 $\boldsymbol{A}$ 是亏损的. 一般说来，亏损矩阵的特征值问题的扰动性态比非亏损矩阵的要复杂得多.

假设 $n\times n$ 矩阵 $\boldsymbol{A}$ 是非亏损的，则相应于特征值 $\lambda_1,\lambda_2,\cdots,\lambda_n$，有 n 个线性无关的特征向量 $\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n$. 令 $\boldsymbol{P}=(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n)$，则 $\boldsymbol{P}$ 非奇异，使

$$\boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{P}=\boldsymbol{D}, \tag{8.1.8}$$

其中 $\boldsymbol{D}=\mathrm{diag}(\lambda_1,\lambda_2,\cdots,\lambda_n)$. 所以说 $\boldsymbol{A}$ 是**可对角化的**(diagonalizable). 这里的 $\boldsymbol{P}$ 是特征向量矩阵，它不是惟一的，它们的集为 $W=\{\boldsymbol{P}\in\mathbb{R}^{n\times n}\,\big|\,\boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{P}=\mathrm{diag}(\lambda_1,\cdots,\lambda_n)\}$.

对于非亏损矩阵 $\boldsymbol{A}$，有扰动 $\delta\boldsymbol{A}$，由此产生的 $\boldsymbol{A}$ 的特征值的扰动有如下定理.

定理 8.1.6(鲍尔-费凯(Bauer-Fike)定理)[10] 设 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$ 是非亏损的，则对任何 $\delta\boldsymbol{A}\in\mathbb{C}^{n\times n}$，$\boldsymbol{A}+\delta\boldsymbol{A}$ 的所有特征值处于 n 个圆盘

$$D_i:\ \{\mu\,\big|\,|\mu-\lambda_i|\leqslant \mathrm{cond}_\nu(\boldsymbol{P})\,\|\delta\boldsymbol{A}\|_\nu\},\quad i=1,2,\cdots,n \tag{8.1.9}$$

的并之中. 其中 $\mathrm{cond}_\nu(\boldsymbol{P})=\|\boldsymbol{P}\|_\nu\,\|\boldsymbol{P}^{-1}\|_\nu$，$\nu=1,2,\infty$.

由定理 8.1.6 可知，$\mathrm{cond}_\nu(\boldsymbol{P})$ 是特征值扰动估计中的放大系数，由于相似变化阵 $\boldsymbol{P}$ 是不惟一的. 所以取 $\mathrm{cond}_\nu(\boldsymbol{P})$ 的下确界

$$\kappa(\boldsymbol{A})=\inf_{\boldsymbol{P}\in W}\{\mathrm{cond}_\nu(\boldsymbol{P})\mid \boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{P}=\mathrm{diag}(\lambda_1,\lambda_2,\cdots,\lambda_n)\}$$

称为特征值问题的**条件数**. 这样有

$$\sigma(\boldsymbol{A}+\delta\boldsymbol{A})\subset\bigcup_{i=1}^{n}\{\mu\in\mathbb{C}\,\big|\,|\mu-\lambda_i|\leqslant\kappa(\boldsymbol{A})\,\|\delta\boldsymbol{A}\|_\nu\}.$$

如果特征向量接近于线性相关(即矩阵是接近亏损的)，则特征值是敏感的. 如果特征向量线性无关，则特征值是不敏感的. 特

别是 $\boldsymbol{A}$ 为正规阵(normal matrix)($\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}=\boldsymbol{A}\boldsymbol{A}^{\mathrm{H}}$),即存在一个酉阵 $\boldsymbol{P}$,使

$$\boldsymbol{P}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{P}=\boldsymbol{D}=\mathrm{diag}(\lambda_1,\lambda_2,\cdots,\lambda_n),$$

其中 $\lambda_i(i=1,\cdots,n)$ 为 $\boldsymbol{A}$ 的特征值,$\boldsymbol{P}$ 是特征向量矩阵,$\boldsymbol{P}^{\mathrm{H}}\boldsymbol{P}=\boldsymbol{I}$. 特征向量是正交的且有 $\mathrm{cond}_2(\boldsymbol{P})=1$,所以正规矩阵(包括实对称阵和复埃尔米特阵)总是良态的. 非正规阵既有对扰动很敏感的特征值,也有不敏感的特征值.

现考虑将扰动理论用于矩阵 $\boldsymbol{A}$ 的单特征值. 设 λ 为 $\boldsymbol{A}$ 的单特征值,$\boldsymbol{x}$ 和 $\boldsymbol{y}$ 分别为相应的右和左特征向量,分别满足 $\boldsymbol{A}\boldsymbol{x}=\lambda\boldsymbol{x}$ 和 $\boldsymbol{y}^{\mathrm{H}}\boldsymbol{A}=\lambda\boldsymbol{y}^{\mathrm{H}}$. 且 $\|\boldsymbol{x}\|_2=1$, $\|\boldsymbol{y}\|_2=1$. 设 $\boldsymbol{A}$ 有扰动 $\delta\boldsymbol{A}$,则

$$(\boldsymbol{A}+\delta\boldsymbol{A})(\boldsymbol{x}+\Delta\boldsymbol{x})=(\lambda+\Delta\lambda)(\boldsymbol{x}+\Delta\boldsymbol{x}).$$

两边展开,忽略二阶小量,然后两端左乘 $\boldsymbol{y}^{\mathrm{H}}$ 且利用 $\boldsymbol{y}^{\mathrm{H}}\boldsymbol{A}=\lambda\boldsymbol{y}^{\mathrm{H}}$,得

$$\boldsymbol{y}^{\mathrm{H}}(\delta\boldsymbol{A})\boldsymbol{x}\approx\Delta\lambda\boldsymbol{y}^{\mathrm{H}}\boldsymbol{x}.$$

因为 λ 是 $\boldsymbol{A}$ 的单特征值,有 $\boldsymbol{y}^{\mathrm{H}}\boldsymbol{x}\neq\boldsymbol{0}$,则得

$$\Delta\lambda\approx\frac{\boldsymbol{y}^{\mathrm{H}}(\delta\boldsymbol{A})\boldsymbol{x}}{\boldsymbol{y}^{\mathrm{H}}\boldsymbol{x}},$$

$$|\Delta\lambda|\leqslant\frac{\|\boldsymbol{y}\|_2\|\boldsymbol{x}\|_2}{|\boldsymbol{y}^{\mathrm{H}}\boldsymbol{x}|}\|\delta\boldsymbol{A}\|_2=\frac{1}{\cos\theta}\|\delta\boldsymbol{A}\|_2,$$

其中 θ 为 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 之间的夹角,$\frac{1}{\cos\theta}$为 $\boldsymbol{A}$ 的单特征值的条件数. 如果右特征向量 $\boldsymbol{x}$ 与左特征向量 $\boldsymbol{y}$ 接近于正交,则单特征值对 δA 的变化是敏感的,反之,$\boldsymbol{x}$ 与 $\boldsymbol{y}$ 接近于平行,它是不敏感的. 事实上,对于实对称阵和复埃尔米特阵,$\boldsymbol{x}=\boldsymbol{y}$,$\cos\theta=1$,所以它们是良态的.

对于重特征值,分析要复杂得多,不过重特征值的 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 有可能是正交的,所以条件数$\frac{1}{\cos\theta}$变得任意大,足以说明重特征值或接近的特征值可能是坏条件的. 特别地,矩阵是亏损的情况.

应当特别引起重视的是矩阵关于特征值问题 $\boldsymbol{A}\boldsymbol{x}=\lambda\boldsymbol{x}$ 的条件数 $k(\boldsymbol{A})$和关于求解方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 的条件数 $\mathrm{cond}(\boldsymbol{A})=\|\boldsymbol{A}\|\times$

$\|A^{-1}\|$是完全不同的两个概念.其中一个是良态的,而另一个可能是严重病态的.如$A=\begin{pmatrix}10^{-10} & 0\\ 0 & 1\end{pmatrix}$,$k_2(A)=1$,$\mathrm{cond}_2(A)=10^{10}$.

8.2 矩阵的变换和矩阵的分解

8.2.1 矩阵的变换和矩阵的舒尔分解

计算特征值与特征向量的许多数值方法基于将原始矩阵A约化为很简单的形式,使得容易求出其特征值和特征向量.如对角阵、三角形矩阵、块三角形矩阵其对角块为1×1和2×2的子阵和上海森伯格阵B(当$i>j+1$时有$b_{ij}=0$).对于前二者其对角线元素就是矩阵A的特征值;对于块三角形矩阵其1×1的元素为A的特征值,其2×2子阵的特征值为A的一对共轭复特征值;对于上海森伯格阵,用QR法求其特征值.

下面给出A可约化为简单形式矩阵的重要定理.

定理8.2.1(舒尔(Schur)定理) 已知$A\in\mathbb{C}^{n\times n}$有特征值$\lambda_1,\cdots,\lambda_n$,它们按任意规定的次序排列,那么存在一个酉矩阵$Q\in\mathbb{C}^{n\times n}$,使得

$$Q^{\mathrm{H}}AQ=R,$$

其中R是具有对角元$r_{jj}=\lambda_j$,$j=1,2,\cdots,n$的上三角阵.

舒尔定理说明任一方阵A酉等价于其对角元依次是A的特征值的三角矩阵,其中$Q^{\mathrm{H}}Q=I$.

设$Q_k=(q_1,q_2,\cdots,q_k)$是酉矩阵Q的前$k(k\leqslant n)$列构成的矩阵,R_k是R的前k维主子阵,则有

$$AQ_k=Q_kR,$$

上式称为A的**部分舒尔分解**.称向量q_i为**舒尔向量**(Schur vectors).称$AQ=QR$为**舒尔分解**(Schur factorization).舒尔向量不是惟一的,它们与矩阵A的特征值的排列次序有关.

定理 8.2.2(实舒尔定理) 设 $A\in\mathbb{R}^{n\times n}$,则存在实正交阵 $Q\in\mathbb{R}^{n\times n}$,使

$$Q^{T}AQ=\begin{pmatrix}\boldsymbol{R}_{11} & \boldsymbol{R}_{12} & \cdots & \boldsymbol{R}_{1m}\\ & \boldsymbol{R}_{22} & \cdots & \boldsymbol{R}_{2m}\\ & & \ddots & \vdots\\ & & & \boldsymbol{R}_{mm}\end{pmatrix},$$

其中每个 $\boldsymbol{R}_{ii}$ 或是 1×1 实矩阵或是具有一对非实的复共轭特征值的 2×2 实矩阵. $\boldsymbol{R}_{ii}(i=1,\cdots,n)$可按任意的次序排列.

定理 8.2.3 如果 $\boldsymbol{A}=(a_{ij})\in\mathbb{C}^{n\times n}$有特征值$\lambda_1,\cdots,\lambda_n$,那么下述命题等价:

(1) $\boldsymbol{A}$ 是正规矩阵,即 $\boldsymbol{A}^{H}\boldsymbol{A}=\boldsymbol{A}\boldsymbol{A}^{H}$.

(2) $\boldsymbol{A}$ 是酉可对角化矩阵,即存在酉阵 $\boldsymbol{Q}$,使 $\boldsymbol{Q}^{H}\boldsymbol{A}\boldsymbol{Q}=\boldsymbol{D}=\mathrm{diag}(\lambda_1,\cdots,\lambda_n)$,$\lambda_i(i=1,\cdots,n)$为 $\boldsymbol{A}$ 的特征值.

(3) $\sum\limits_{i,j=1}^{n}|a_{ij}|^{2}=\sum\limits_{j=1}^{n}|\lambda_j|^{2}$.

(4) 存在由 $\boldsymbol{A}$ 的 n 个特征向量组成的标准正交基.

在该定理中,(1)与(2)的等价性常常被认为正规矩阵的谱定理.也说明正规矩阵是可对角化的.酉矩阵、实对称阵、复埃尔米特阵 $\boldsymbol{A}^{H}=\boldsymbol{A}$、斜埃尔米特阵 $\boldsymbol{A}^{H}=-\boldsymbol{A}$ 及斜实对称阵 $\boldsymbol{A}^{T}=-\boldsymbol{A}$ 等都是正规阵.

8.2.2 豪斯霍尔德变换和吉文斯变换

豪斯霍尔德变换和吉文斯(Givens)变换是矩阵计算的有力工具.在7.12.2节用豪斯霍尔德变换构造克雷洛夫子空间的一组标准正交基.在 7.12.3 节用吉文斯变换将上海森伯格阵变换为上三角阵.在本章将用它们将矩阵约化为上海森伯格阵和对矩阵作 QR 分解.这里主要讨论实矩阵和实向量,易推广到复的情况.

定义 8.2.4 设 $w\in\mathbb{R}^n$ 且 $\|w\|_2=1$,则初等矩阵

$$P=I-2ww^{\mathrm{T}}$$

称为**豪斯霍尔德变换**或称**豪斯霍尔德变换阵**.

豪斯霍尔德变换阵是对称正交阵,即 $P=P^{\mathrm{T}}$ 且 $P^{\mathrm{T}}P=I$. 由 $y=Px$ 得到的 y,其 2 范数 $\|y\|_2=\|x\|_2$.

豪斯霍尔德变换的应用之一是:对任意 $x\in\mathbb{R}^n$,$x=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}$,都可以构造豪斯霍尔德变换阵 P,使得

$$Px=\sigma e_1,$$

其中 $|\sigma|=\|x\|_2$. 求矩阵 P 的计算公式如下:

$$\begin{cases}\sigma=-\operatorname{sgn}(x_1)\|x\|_2,\\ \operatorname{sgn}(x_1)=\begin{cases}1, & 当\ x_1\geqslant 0,\\ -1, & 当\ x_1<0,\end{cases}\\ u=x-\sigma e_1=(x_1+\operatorname{sgn}(x_1)\|x\|_2,x_2,\cdots,x_n)^{\mathrm{T}},\\ \|u\|_2^2=2\|x\|_2(\|x\|_2+|x_1|),\\ P=I-\dfrac{2uu^{\mathrm{T}}}{\|u\|_2^2}.\end{cases}\tag{8.2.1}$$

类似地,若豪斯霍尔德变换可将 x 的第 $j+2$ 到 k 个分量变换为 0,则豪斯霍尔德变换阵 P 的计算公式改为:

$$\begin{cases}\alpha=(x_{j+1}^2+\cdots+x_k^2)^{\frac{1}{2}},\\ \beta=-\operatorname{sgn}(x_{j+1})\alpha,\\ u=(0,\cdots,0,x_{j+1}+\operatorname{sgn}(x_{j+1})\alpha,x_{j+2},\cdots,x_k,0,\cdots,0)^{\mathrm{T}},\\ \|u\|_2^2=2\alpha(\alpha+|x_{j+1}|),\\ P=I-\dfrac{2uu^{T}}{\|u\|_2^2}=\begin{pmatrix}I_j & & \\ & \bar{P} & \\ & & I_{n-k}\end{pmatrix},\end{cases}\tag{8.2.2}$$

其中 $\bar{P}\in\mathbb{R}^{(k-j)\times(k-j)}$. 得

$$\boldsymbol{Px}=(x_1,x_2,\cdots,x_j,\beta,0,\cdots,0,x_{k+1},\cdots,x_n)^{\mathrm{T}} \tag{8.2.3}$$

当 $k=n$ 时,有

$$\boldsymbol{Px}=(x_1,x_2,\cdots,x_j,\beta,0,\cdots,0)^{\mathrm{T}}. \tag{8.2.4}$$

定义 8.2.5 对某个 θ,记 $s=\sin\theta, c=\cos\theta$,并设 $1\leqslant i\leqslant k\leqslant n$,则称正交矩阵

$$\boldsymbol{J}(i,k,\theta)=\begin{pmatrix} 1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 1 & & & & & & \\ & & & c & & s & & & \\ & & & & 1 & & & & \\ & & & & & \ddots & & & \\ & & & & & 1 & & & \\ & & & -s & & c & & & \\ & & & & & & 1 & & \\ & & & & & & & \ddots & \\ & & & & & & & & 1 \end{pmatrix}\begin{matrix} \\ \\ \\ i \\ \\ \\ \\ k \\ \\ \\ \\ \end{matrix} \tag{8.2.5}$$

为吉文斯变换或吉文斯变换阵或平面旋转矩阵.

定理 8.2.6 设 $\boldsymbol{x}\in\mathbb{R}^n$ 的第 k 个分量 $x_k\neq 0, 1\leqslant i\leqslant k\leqslant n$. 若令

$$c=\cos\theta=\frac{x_i}{\sqrt{x_i^2+x_k^2}},\quad s=\sin\theta=\frac{x_k}{\sqrt{x_i^2+x_k^2}},$$

则 $\boldsymbol{y}=\boldsymbol{J}(i,k,\theta)\boldsymbol{x}$ 的分量为

$$\begin{cases} y_i=\sqrt{x_i^2+x_k^2}, & y_k=0, \\ y_j=x_j, & j\neq i,k. \end{cases}$$

它的意义是消去向量 $\boldsymbol{x}$ 的第 k 个非零元素,改变了第 i 个元素,其余元素不变. $\boldsymbol{J}(i,k,\theta)$ 是正交阵,即 $\boldsymbol{J}^{\mathrm{T}}(i,k,\theta)\boldsymbol{J}(i,k,\theta)=\boldsymbol{I}$. $\boldsymbol{J}(i,k,\theta)\boldsymbol{A}$ 只改变 $\boldsymbol{A}$ 的第 i,k 行. 同理, $\boldsymbol{AJ}(i,k,\theta)$,只改变 $\boldsymbol{A}$ 的第 i,k 列. 正交相似变换 $\boldsymbol{J}^{\mathrm{T}}(i,k,\theta)\boldsymbol{AJ}(i,k,\theta)$ 只改变 $\boldsymbol{A}$ 的第 i,k 行和 i,k 列.

8.2.3 化矩阵为海森伯格形

定义 8.2.7 若矩阵 $\boldsymbol{B}=(b_{ij})\in\mathbb{R}^{n\times n}$ 的次对角线以下元素为零,即

$$b_{ij}=0,\quad i>j+1,$$

即 $\boldsymbol{B}$ 的形状为

$$\boldsymbol{B}=\begin{pmatrix} b_{11} & b_{12} & b_{13} & \cdots & b_{1n} \\ b_{21} & b_{22} & b_{23} & \cdots & b_{2n} \\ & b_{32} & b_{33} & \cdots & b_{3n} \\ & & \ddots & \ddots & \vdots \\ & & & b_{n,n-1} & b_{nn} \end{pmatrix}, \tag{8.2.6}$$

则称 $\boldsymbol{B}$ 为(上)**海森伯格阵**,简称**海森伯格形**.若某个次对角元为零,如

$$b_{k+1,k}=0,\quad 1\leqslant k\leqslant n-1,$$

则说 $\boldsymbol{B}$ **是可约的**,否则称 $\boldsymbol{B}$ **是不可约的**.

可约的海森伯格形 $\boldsymbol{B}$ 可划分为分块上三角形,则 $\boldsymbol{B}$ 的全部特征值是各对角子方块阵的特征值的集合.因此,可把原特征值问题降为低阶特征值问题.

定理 8.2.8 对任意矩阵 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$,存在正交阵 $\boldsymbol{Q}$,使得

$$\boldsymbol{B}=\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}$$

为(上)海森伯格阵.即任何方阵均可通过正交相似变换为海森伯格阵.

推论 8.2.9 对任意对称矩阵 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$,存在正交阵 $\boldsymbol{Q}$,使得 $\boldsymbol{B}=\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}$ 为对称的三对角阵.

豪斯霍尔德变换阵和吉文斯变换阵都是正交阵,均可将任意矩阵 $\boldsymbol{A}$ 相似变换为(上)海森伯格阵.

若用豪斯霍尔德变换对 $\boldsymbol{A}$ 作逐次正交相似变换,可逐次地把 $\boldsymbol{A}$ 每列的次对角线以下(不包括次对角线)的元素消为零,且每次变换后 $\boldsymbol{A}$ 的特征值保持不变,这样,可简化求解特征值问题.

算法 8.2.10 豪斯霍尔德正交相似变换化 $\boldsymbol{A}$ 为上海森伯格阵

(1) 对于 $j=1,\cdots,n-2$

(2) 定义向量 $\boldsymbol{x}=\boldsymbol{a}_{*,j}$(矩阵 $\boldsymbol{A}$ 的第 j 列元素)

(3) 按公式(8.2.2)计算豪斯霍尔德阵 $\boldsymbol{P}_j$,其中 $k=n$

(4) 正交相似变换 $\boldsymbol{P}_j\boldsymbol{A}\boldsymbol{P}_j\Rightarrow\boldsymbol{A}$

(5) j 尾

算法的每一步执行了一次正交相似变换,变换后的矩阵仍存放在 $\boldsymbol{A}$ 中. 最后 $\boldsymbol{A}$ 呈上海森伯格形. 即 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}=\boldsymbol{B}$,其中 $\boldsymbol{Q}=\boldsymbol{P}_1\boldsymbol{P}_2\cdots\boldsymbol{P}_{n-2}$.

例 8.2.11 用豪斯霍尔德正交相似变换化矩阵

$$\boldsymbol{A}=\begin{pmatrix}1 & 2 & 1 & 2\\ 2 & 2 & -1 & 1\\ 1 & -1 & 1 & 1\\ 2 & 1 & 1 & 1\end{pmatrix}$$

为三对角阵.

解 $j=1,\boldsymbol{x}=a_{*,1},\alpha=(a_{21}^2+a_{31}^2+a_{41}^2)^{\frac{1}{2}}=3,\boldsymbol{u}=(0,5,1,2)^{\mathrm{T}}$, $\|\boldsymbol{u}\|_2^2=30$,

$$\boldsymbol{P}_1=\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & -0.6667 & -0.3333 & -0.6667\\ 0 & -0.3333 & 0.9333 & -0.1333\\ 0 & -0.6667 & -0.1333 & 0.7333\end{pmatrix},$$

$$\boldsymbol{A}_1=\boldsymbol{P}_1\boldsymbol{A}\boldsymbol{P}_1=\begin{pmatrix}1 & -3 & 0 & 0\\ -3 & 2.333 & 0.4666 & -0.6667\\ 0 & 0.4666 & 1.5733 & 1.3467\\ 0 & -0.06667 & 1.3467 & 0.4667\end{pmatrix}\Rightarrow\boldsymbol{A},$$

$j=2$, $\boldsymbol{x}=a_{*,2}$, $\alpha=(a_{32}^2+a_{42}^2)=0.4714$, $\boldsymbol{u}=(0,0,0.9381,-0.06667)^{\mathrm{T}}$, $\|\boldsymbol{u}\|_2^2=0.8845$,

$$
\boldsymbol{P}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.8800 & -0.06254 \\ 0 & 0 & -0.06254 & 0.004445 \end{bmatrix},
$$

$$
\boldsymbol{A}_2 = \boldsymbol{P}_2\boldsymbol{A}_1\boldsymbol{P}_2
= \begin{bmatrix} 1 & -3 & 0 & 0 \\ -3 & 2.333 & -0.4714 & 0 \\ 0 & -0.4714 & 1.167 & -1.500 \\ 0 & 0 & -1.500 & 0.5000 \end{bmatrix}.
$$

用吉文斯变换对矩阵 $\boldsymbol{A}$ 作一系列的正交相似变换，也可将 $\boldsymbol{A}$ 化为海森伯格形，只是每次变换仅消去对角线元素之下的一个元素. 显然增加了计算量.

8.2.4 矩阵的 QR 分解

定理 8.2.12（QR 分解定理） 设 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 非奇异，则存在正交阵 $\boldsymbol{Q}$ 与上三角阵 $\boldsymbol{R}$，使 $\boldsymbol{A}$ 有如下分解

$$
\boldsymbol{A} = \boldsymbol{QR},
$$

且当 $\boldsymbol{R}$ 的对角元均为正时，分解是惟一的，否则不是惟一的.

豪斯霍尔德变换和吉文斯变换均可用来对矩阵作 QR 分解. 由于上海森伯格阵在矩阵特征值问题的 QR 算法中有重要的意义. 这里重点讨论上海森伯格阵的 QR 分解，吉文斯变换是最合适的选择.

用一系列的吉文斯变换阵 $\boldsymbol{J}(j,j+1,\theta_j)$，$j=1,2,\cdots,n-1$ 左乘形如式(8.2.6)的上海森伯格阵 $\boldsymbol{B}$，依次将 $\boldsymbol{B}$ 的对角线以下的仅有的一个非零元素消为零，得到上三角阵 $\boldsymbol{R}$，其中

$$
\boldsymbol{A}=\begin{bmatrix}4 & 4 & 0\\ 3 & 3 & -1\\ 0 & 1 & 1\end{bmatrix}
$$

的 QR 分解，使 $\boldsymbol{R}$ 的对角线元素为正.

解

$$
\boldsymbol{J}_1=\begin{bmatrix}\frac{4}{5} & \frac{3}{5} & 0\\ -\frac{3}{5} & \frac{4}{5} & 0\\ 0 & 0 & 1\end{bmatrix},\quad \boldsymbol{J}_1\boldsymbol{A}=\begin{bmatrix}5 & 5 & -\frac{3}{5}\\ 0 & 0 & -\frac{4}{5}\\ 0 & 1 & 1\end{bmatrix},
$$

$$
\boldsymbol{J}_2=\begin{bmatrix}1 & 0 & 0\\ 0 & 0 & 1\\ 0 & -1 & 0\end{bmatrix},\quad \boldsymbol{J}_2(\boldsymbol{J}_1\boldsymbol{A})=\begin{bmatrix}5 & 5 & -\frac{3}{5}\\ 0 & 1 & 1\\ 0 & 0 & \frac{4}{5}\end{bmatrix}=\boldsymbol{R},
$$

$$
\boldsymbol{Q}=\boldsymbol{J}_1^{\mathrm{T}}\boldsymbol{J}_2^{\mathrm{T}}=\begin{bmatrix}\frac{4}{5} & 0 & \frac{3}{5}\\ \frac{3}{5} & 0 & -\frac{4}{5}\\ 0 & 1 & 0\end{bmatrix},
$$

易验证 $\boldsymbol{A}=\boldsymbol{Q}\boldsymbol{R}$.

如果 $\boldsymbol{A}=\boldsymbol{Q}\boldsymbol{R}$ 是 $\boldsymbol{A}$ 的 QR 分解，其中 $\boldsymbol{A}\in\mathbb{R}^{n\times m}$ 且 $n>m$，$\boldsymbol{Q}\in\mathbb{R}^{n\times m}$ 和 $\boldsymbol{R}\in\mathbb{R}^{m\times m}$，称此分解为**缩减**(reduced)**QR 分解**.

定理 8.2.15 假定 $\boldsymbol{A}\in\mathbb{R}^{n\times m}$ 是列满秩的，则缩减 QR 分解

$$
\boldsymbol{A}=\boldsymbol{Q}_1\boldsymbol{R}_1
$$

是惟一的，其中 $\boldsymbol{Q}_1\in\mathbb{R}^{n\times m}$ 的列向量相互正交，$\boldsymbol{R}_1$ 是对角线元素大于 0 的上三角阵. 而且 $\boldsymbol{R}_1=\boldsymbol{G}^{\mathrm{T}}$，其中 $\boldsymbol{G}$ 是 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ 的下三角楚列斯基因子.

8.3 幂法和反幂法

8.3.1 幂法

设 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 可对角化,即存在可逆阵 $\boldsymbol{X}$,使 $\boldsymbol{X}^{-1}\boldsymbol{A}\boldsymbol{X}=\mathrm{diag}(\lambda_1,\lambda_2,\cdots,\lambda_n)$,其中 $\boldsymbol{X}=(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n)$,$\{\boldsymbol{x}_i\}_{i=1}^n$ 是线性无关的. 又设

$$|\lambda_1|>|\lambda_2|\geqslant\cdots\geqslant|\lambda_n|$$

且满足

$$\boldsymbol{A}\boldsymbol{x}_i=\lambda_i\boldsymbol{x}_i,\quad i=1,2,\cdots,n,$$

其中按模最大的特征值 λ_1 称为**主特征值**. 它显然是非零单实的;对应的特征向量 $\boldsymbol{x}_1$ 称为**主特征向量**. **幂法**(power method)是用于求矩阵 $\boldsymbol{A}$ 的主特征值和主特征向量的迭代法.

算法 8.3.1　幂法

(1) 任给 $\boldsymbol{x}^{(0)}\in\mathbb{R}^n$ 和给定误差界 $\varepsilon>0$

(2) 对于 $k=1,2,\cdots$

(3) 　　$\boldsymbol{y}^{(k)}=\boldsymbol{A}\boldsymbol{x}^{(k-1)}$

(4) 　　$m_k=\max(\boldsymbol{y}^{(k)})$

(5) 　　$\boldsymbol{x}^{(k)}=\dfrac{\boldsymbol{y}^{(k)}}{m_k}$

(6) 　　如果 $\dfrac{|m_k-m_{k-1}|}{|m_k|}<\varepsilon$,则终止,否则

(7) 　　k 尾

上述算法中,$\max(\boldsymbol{y}^{(k)})$,表示 $\boldsymbol{y}^{(k)}$ 的按模最大的分量. 对任意 $\boldsymbol{x}^{(0)}\in\mathbb{R}^n$ 有

$$\boldsymbol{x}^{(0)}=\sum_{i=1}^n\alpha_i\boldsymbol{x}_i.$$

只要 $\alpha_1\neq0$,则算法 8.3.1 生成的迭代序列具有收敛性质:

$$\lim_{k\to\infty}\boldsymbol{x}^{(k)}=\boldsymbol{x}_1/\max(\boldsymbol{x}_1),\lim_{k\to\infty}m_k=\lambda_1$$

$$J_j = J(j, j+1, \theta_j) = \begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & & c & s & & \\ & & & -s & c & & \\ & & & & & 1 & \\ & & & & & & \ddots \\ & & & & & & & 1 \end{pmatrix} \begin{matrix} \\ \\ \\ j \\ j+1 \\ \\ \\ \\ \end{matrix}.$$

(8.2.7)

算法 8.2.13 上海森伯格阵 $\boldsymbol{B}$ 的 QR 分解

(1) $\boldsymbol{S}=\boldsymbol{I}\in\mathbb{R}^{n\times n}$

(2) 对于 $j=1,2,\cdots,n-1$

(3) 　　计算形如式(8.2.7)的吉文斯变换阵 $\boldsymbol{J}_j$，其中

(4) 　　$c:=\dfrac{b_{jj}}{\sqrt{b_{jj}^2+b_{j+1,j}^2}}, s:=\dfrac{b_{j+1,j}}{\sqrt{b_{jj}^2+b_{j+1,j}^2}}$

(5) 　　$\boldsymbol{B}:=\boldsymbol{J}_j\boldsymbol{B}$

(6) 　　$\boldsymbol{S}:=\boldsymbol{J}_j\boldsymbol{S}$

(7) 　　j 尾

(8) $\boldsymbol{Q}=\boldsymbol{S}^{\mathrm{T}}$

完成上述算法之后，

$$\boldsymbol{J}_{n-1}\boldsymbol{J}_{n-2}\cdots\boldsymbol{J}_1\boldsymbol{B}=\boldsymbol{R},$$

其中 $\boldsymbol{R}$ 为上三角阵，保存在 $\boldsymbol{B}$ 内. $\boldsymbol{S}=\boldsymbol{J}_{n-1}\boldsymbol{J}_{n-2}\cdots\boldsymbol{J}_1$，

$$\boldsymbol{Q}=\boldsymbol{J}_1^{\mathrm{T}}\boldsymbol{J}_2^{\mathrm{T}}\cdots\boldsymbol{J}_{n-1}^{\mathrm{T}}$$

为正交阵，则

$$\boldsymbol{B}=\boldsymbol{Q}\boldsymbol{R}$$

给出了上海森伯格阵 $\boldsymbol{B}$ 的 QR 分解.

例 8.2.14 用吉文斯变换，求矩阵

且 m_k 的收敛速度决定于 $\left|\frac{\lambda_2}{\lambda_1}\right|$，当 k 大时，有

$$|m_k-\lambda_1|\approx c\left|\frac{\lambda_2}{\lambda_1}\right|^k, \quad c\text{ 为常数}. \tag{8.3.1}$$

$$\frac{|m_{k+1}-\lambda_1|}{|m_k-\lambda_1|}\approx\left|\frac{\lambda_2}{\lambda_1}\right|. \tag{8.3.2}$$

说明对于大型稀疏矩阵且 $|\lambda_1|\gg|\lambda_2|$ 时，幂法收敛快. 但当 $|\lambda_1|\approx|\lambda_2|$ 时，收敛很慢，需要加速.

当可对角化矩阵 $\boldsymbol{A}$ 有重特征值，即

$$\lambda_1=\lambda_2=\cdots=\lambda_r,\ |\lambda_r|\geqslant|\lambda_{r+1}|\geqslant\cdots\geqslant|\lambda_n|$$

时，如果 $\boldsymbol{x}^{(0)}=\sum_{i=1}^{n}\alpha_i\boldsymbol{x}_i$ 的系数 $\alpha_1,\alpha_2,\cdots,\alpha_r$ 不全为零，则算法 8.3.1 生成的迭代序列具有收敛性：

$$\lim_{k\to\infty}\boldsymbol{x}^{(k)}=\sum_{i=1}^{r}\alpha_i\boldsymbol{x}_i/\max\left(\sum_{i=1}^{r}\alpha_i\boldsymbol{x}_i\right),\quad \lim_{k\to\infty}m_k=\lambda_1$$

收敛速度取决于 $|\lambda_{r+1}/\lambda_1|$.

当 $\boldsymbol{A}$ 亏损时，幂法的收敛速度非常慢.

8.3.2 加速方法

1. 艾特肯加速法

由式(8.3.2)知，幂法生成的序列 $m_k(k=1,2,\cdots)$ 是线性收敛于 λ_1 的. 因此，对于充分大的 k，有

$$\frac{m_{k+2}-\lambda_1}{m_{k+1}-\lambda_1}\approx\frac{m_{k+1}-\lambda_1}{m_k-\lambda_1},$$

求得 $\lambda_1\approx\widetilde{m}_k$，

$$\widetilde{m}_k=\frac{m_km_{k+2}-m_{k+1}^2}{m_{k+2}-2m_{k+1}+m_k}. \tag{8.3.3}$$

同理，可求得特征向量的新估计 $\widetilde{\boldsymbol{X}}^{(k)}$，其分量有

$$(\widetilde{\boldsymbol{x}}^{(k)})_i=\frac{(\boldsymbol{x}^{(k)})_i(\boldsymbol{x}^{(k+2)})_i-(\boldsymbol{x}^{(k+1)})_i^2}{(\boldsymbol{x}^{(k+2)})_i-2(\boldsymbol{x}^{(k+1)})_i+(\boldsymbol{x}^{(k)})_i},\quad i=1,2,\cdots,n. \tag{8.3.4}$$

公式(8.3.3)和(8.3.4)提高了幂法的收敛速度.称这种加速法为**埃特肯**(Aitken)**外推加速法**.

2. 瑞利商加速

设 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 是对称阵,对于任一非零向量 $\boldsymbol{x}$,称

$$R(\boldsymbol{x})=\frac{(\boldsymbol{Ax},\boldsymbol{x})}{(\boldsymbol{x},\boldsymbol{x})} \tag{8.3.5}$$

为对应于向量 $\boldsymbol{x}$ 的**瑞利**(Rayleigh)**商**.应用瑞利商于幂法,有如下算法.

算法 8.3.2 瑞利商加速幂法

(1) 任给 $\boldsymbol{x}^{(0)}\in\mathbb{R}^n$,给定误差界 $\varepsilon>0$

(2) 对于 $k=1,2,\cdots$

(3) $\quad\boldsymbol{y}^{(k)}=\boldsymbol{Ax}^{(k-1)}$

(4) $\quad m_k=\dfrac{(\boldsymbol{Ax}^{(k-1)},\boldsymbol{x}^{(k-1)})}{(\boldsymbol{x}^{(k-1)},\boldsymbol{x}^{(k-1)})}$

(5) $\quad\boldsymbol{x}^{(k)}=\boldsymbol{y}^{(k)}/m_k$

(6) $\quad$如果 $\dfrac{|m_k-m_{k-1}|}{|m_k|}<\varepsilon$,则终止,否则

(7) $\quad k$ 尾

特别地,当瑞利商中的 $(\boldsymbol{x},\boldsymbol{x})=\|\boldsymbol{x}\|_2^2=1$ 时,$R(\boldsymbol{x})=(\boldsymbol{Ax},\boldsymbol{x})$.如果上述算法中,$\boldsymbol{x}^{(k)}$ 用欧氏范数规范化,则有修正算法.

算法 8.3.3 修正瑞利商加速幂法

(1) 给定误差界 $\varepsilon>0$ 和 $\boldsymbol{x}^{(0)}\in\mathbb{R}^n$ 使 $\|\boldsymbol{x}^{(0)}\|_2=1$,如 $\boldsymbol{x}^{(0)}=\dfrac{1}{\sqrt{n}}(1,\cdots,1)^{\mathrm{T}}$

(2) 对于 $k=1,2,\cdots$

(3) $\quad\boldsymbol{y}^{(k)}=\boldsymbol{Ax}^{(k-1)}$

(4) $\quad m_k=(\boldsymbol{Ax}^{(k-1)},\boldsymbol{x}^{(k-1)})=(\boldsymbol{y}^{(k)})^{\mathrm{T}}\boldsymbol{x}^{(k-1)}$

(5) $\quad\boldsymbol{x}^{(k)}=\boldsymbol{y}^{(k)}/\|\boldsymbol{y}^{(k)}\|_2$

(6) 如果$\frac{|m_k-m_{k-1}|}{|m_k|}<\varepsilon$,则终止,否则

(7) k 尾

定理 8.3.4 设 $A\in\mathbb{R}^{n\times n}$为对称阵,特征值满足$|\lambda_1|>|\lambda_2|\geqslant\cdots\geqslant|\lambda_n|$;对应的特征向量满足$(x_i,x_j)=\delta_{ij}$,则瑞利商加速幂法计算 λ_1 得到的近似 m_k,有如下估计

$$R(x^{(k)})=\lambda_1+O\left(\left(\frac{\lambda_2}{\lambda_1}\right)^{2k}\right). \tag{8.3.6}$$

对于幂法的迭代序列,有估计式(8.3.1),即

$$m_k=\lambda_1+O\left(\left(\frac{\lambda_2}{\lambda_1}\right)^{k}\right).$$

显然,瑞利商逼近 λ_1 的速度比幂法快,提高了幂法的收敛速度.

8.3.3 收缩方法

用幂法可求出可对角化的矩阵 A 的主特征值,但不能求其他特征值.现设$|\lambda_1|>|\lambda_2|>|\lambda_3|\geqslant\cdots\geqslant|\lambda_n|$.在求出 λ_1 之后,用收缩的方法求 λ_2.

收缩方法很多,这里仅讨论**舒尔-维兰特**(Schur-Weilandt)收缩法和以相似变换为基础的收敛法.

1. 舒尔-维兰特收缩法

维兰特(Weilandt)收缩法的收缩矩阵为

$$A_1=A-\sigma x_1 v^{H}, \tag{8.3.7}$$

其中v为满足$v^{H}x_1=1$的任意向量,σ为可选择的参数,则 A_1 的特征值除了由 A 的特征值λ_1 变为 $\lambda_1-\sigma$ 之外,其余的与 A 的特征值相同.

定理 8.3.5(维兰特定理) 由式(8.3.7)定义的 A_1,其谱是

$$\sigma(A_1)=\{\lambda_1-\sigma,\lambda_2,\lambda_3,\cdots,\lambda_n\}.$$

式(8.3.7)中的向量v有许多种选择方法.若取$v=w_1$,w_1 为

$\boldsymbol{A}$ 的第一个左特征向量，称此方法为**霍特林(Hotelling)收缩法**. 另一种简单的选择：就取 $\boldsymbol{v}=\boldsymbol{x}_1$. 可以证明选 $\boldsymbol{v}=\boldsymbol{x}_1$ 是最优的.

命题 8.3.6 设 λ_1 和 x_1 是 $\boldsymbol{A}$ 的特征值和相应的特征向量，且 $\|\boldsymbol{x}_1\|_2=1$，令 $\boldsymbol{A}_1=\boldsymbol{A}-\sigma\boldsymbol{x}_1\boldsymbol{x}_1^{\mathrm{H}}$，则 $\boldsymbol{A}_1$ 的特征值是 $\tilde{\lambda}_1=\lambda_1-\sigma$，$\tilde{\lambda}_j=\lambda_j$，$j=2,3,\cdots,n$，而且相应于 $\tilde{\lambda}_j(j=1,2,\cdots,n)$ 的舒尔向量与 $\boldsymbol{A}$ 的相同.

证明 设 $\boldsymbol{AU}=\boldsymbol{UR}$ 为 $\boldsymbol{A}$ 的舒尔分解，其中 $\boldsymbol{R}$ 是上三角阵，$\boldsymbol{U}$ 是正交规范化的，则有

$$\boldsymbol{A}_1\boldsymbol{U}=(\boldsymbol{A}-\sigma\boldsymbol{x}_1\boldsymbol{x}_1^{\mathrm{H}})\boldsymbol{U}=\boldsymbol{UR}-\sigma\boldsymbol{x}_1\boldsymbol{e}_1^{\mathrm{H}}=\boldsymbol{U}(\boldsymbol{R}-\sigma\boldsymbol{e}_1\boldsymbol{e}_1^{\mathrm{H}}).$$

得证.

可反复用维兰特收缩法，令

$$\boldsymbol{A}_i=\boldsymbol{A}_{i-1}-\sigma_{i-1}\boldsymbol{q}_{i-1}\boldsymbol{q}_{i-1}^{\mathrm{h}}. \tag{8.3.8}$$

在用幂法求出 $\boldsymbol{A}$ 的主特征值 λ_1 和主特征向量 $\boldsymbol{x}_1$ 后，取 $\sigma_1=\lambda_1$，$q_1=\dfrac{\boldsymbol{x}_1}{\|\boldsymbol{x}_1\|_2}$. 由式(8.3.8)得 $\boldsymbol{A}_1$，用幂法于 $\boldsymbol{A}_1$，得 λ_2 和 $\tilde{x}_2$ 后，取 $\sigma_2=\lambda_2$，对 q_1，正交规范化 $\tilde{\boldsymbol{x}}_2$，得 q_2，继续收缩，可依次得到 λ_1，$\lambda_2,\cdots,\lambda_j$ 及舒尔向量 $\boldsymbol{q}_1,\boldsymbol{q}_2,\cdots,\boldsymbol{q}_j$，从而构成**舒尔-维兰特收缩法**.

算法 8.3.7 舒尔-维兰特收缩法

(1) 定义 $\boldsymbol{A}_1\equiv\boldsymbol{A}$，求出 λ_1 和 $\boldsymbol{x}_1$，计算 $q_1=\dfrac{\boldsymbol{x}_1}{\|\boldsymbol{x}_1\|_2}$，取 $\sigma=\lambda_1$

(2) 对于 $i=2,3,\cdots,j$

(3) 　　定义 $\boldsymbol{A}_i=\boldsymbol{A}_{i-1}-\sigma\boldsymbol{q}_{i-1}\boldsymbol{q}_{i-1}^{\mathrm{H}}$

(4) 　　计算 $\boldsymbol{A}_i$ 的主特征值 λ_i 和相应的特征向量 $\hat{\boldsymbol{x}}_i$

(5) 　　$\sigma=\lambda_i$

(6) 　　对 $\boldsymbol{q}_1,\boldsymbol{q}_2,\cdots,\boldsymbol{q}_{j-1}$ 正交规范化 $\hat{\boldsymbol{x}}_i$，得 $\boldsymbol{q}_i$

(7) 　　i 尾

令 $\boldsymbol{Q}_j\equiv(\boldsymbol{q}_1,\boldsymbol{q}_2,\cdots,\boldsymbol{q}_j)$，其中 $\boldsymbol{q}_1,\boldsymbol{q}_2,\cdots,\boldsymbol{q}_j$ 为对应于特征值 $\lambda_1,\cdots,\lambda_j$ 的舒尔向量，$\boldsymbol{Q}_j$ 为正交阵.

命题 8.3.8 设 $\boldsymbol{\Sigma}_j=\text{diag}(\sigma_1,\sigma_2,\cdots,\sigma_j)$，$\boldsymbol{Q}_j$ 是相应于 $\lambda_1,\cdots,\lambda_j$ 的 $\boldsymbol{A}$ 的舒尔向量组成的 $n\times j$ 正交阵，则矩阵

$$\boldsymbol{A}_j\equiv\boldsymbol{A}-\boldsymbol{Q}_j\boldsymbol{\Sigma}_j\boldsymbol{Q}_j^{\mathrm{H}}$$

的特征值 $\tilde{\lambda}_i=\lambda_i-\sigma_i,i\leqslant j,\tilde{\lambda}_i=\lambda_i,i>j$. 其相应的舒尔向量与 $\boldsymbol{A}$ 的相同.

2. 基于相似变换的收缩法

设 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$，当用幂法求得 $\boldsymbol{A}$ 的主特征值 λ_1 和主特征向量 $\boldsymbol{x}_1$ 时，x_1 已知时，可找到一个豪斯霍尔德变换阵 $\boldsymbol{P}_1$，使

$$\boldsymbol{P}_1\boldsymbol{x}_1=k_1\boldsymbol{e}_1,\tag{8.3.9}$$

其中 $k_1\neq0$. 由 $\boldsymbol{A}\boldsymbol{x}_1=\lambda_1\boldsymbol{x}_1$，得

$$\boldsymbol{P}_1\boldsymbol{A}\boldsymbol{P}_1^{\mathrm{H}}\boldsymbol{e}_1=\lambda_1\boldsymbol{e}_1.$$

记

$$\boldsymbol{A}_2=\boldsymbol{P}_1\boldsymbol{A}\boldsymbol{P}_1^{\mathrm{H}}=\begin{pmatrix}\lambda_1 & \boldsymbol{b}_1^{\mathrm{T}}\\ \boldsymbol{0} & \boldsymbol{B}_2\end{pmatrix}\tag{8.3.10}$$

其中 $\boldsymbol{B}_2\in\mathbb{C}^{(n-1)\times(n-1)}$. 显然 $\boldsymbol{B}_2$ 的特征值是 $\lambda_2,\cdots,\lambda_n$. 可用幂法求 $\boldsymbol{B}_2$ 的主特征值 λ_2 和主特征向量 $\boldsymbol{y}_2\in\mathbb{C}^{n-1}$，即

$$\boldsymbol{B}_2\boldsymbol{y}_2=\lambda_2\boldsymbol{y}_2.$$

设 $\boldsymbol{A}_2\boldsymbol{z}_2=\lambda_2\boldsymbol{z}_2$，其中 $\boldsymbol{z}_2=(\alpha,\boldsymbol{y}^{\mathrm{T}})^{\mathrm{T}}\in\mathbb{C}^n$，$\alpha$ 为待定常数，则有

$$\boldsymbol{x}_2=\boldsymbol{P}_1^{\mathrm{H}}\boldsymbol{z}_2,\tag{8.3.11}$$

$$\begin{cases}(\lambda_2-\lambda_1)\alpha=\boldsymbol{b}_1^{\mathrm{T}}\boldsymbol{y}\\ \boldsymbol{B}_2\boldsymbol{y}=\lambda_2\boldsymbol{y}.\end{cases}$$

因为 $\lambda_1\neq\lambda_2$，可取 $\boldsymbol{y}=\boldsymbol{y}_2,\alpha=\dfrac{\boldsymbol{b}_1^{\mathrm{T}}\boldsymbol{y}_2}{\lambda_2-\lambda_1}$，求得 $\boldsymbol{z}_2$. 由式(8.3.11)可得 $\boldsymbol{A}$ 对应于 λ_2 的特征向量 $\boldsymbol{x}_2$.

如果 $\boldsymbol{A}$ 的特征值相互隔离较好，则用类似的方法再对 $\boldsymbol{B}_2$ 进

行收缩，设$\hat{\boldsymbol{P}}_2\in\mathbb{C}^{(n-1)\times(n-1)}$满足

$$\hat{\boldsymbol{P}}_2\boldsymbol{y}_2=k_2\boldsymbol{e}_1,\boldsymbol{e}_1\in\mathbb{R}^{(n-1)},$$

则有

$$\hat{\boldsymbol{P}}_2\boldsymbol{B}_2\hat{\boldsymbol{P}}_2^{\mathrm{H}}=\begin{pmatrix}\lambda_2 & \boldsymbol{b}_2^{\mathrm{T}}\\ \boldsymbol{0} & \boldsymbol{B}_3\end{pmatrix}.$$

令 $\boldsymbol{P}_2=\begin{pmatrix}1 & \boldsymbol{0}\\ \boldsymbol{0} & \hat{\boldsymbol{P}}_2\end{pmatrix}$，则

$$\boldsymbol{A}_3=\boldsymbol{P}_2\boldsymbol{P}_1\boldsymbol{A}\boldsymbol{P}_1^{\mathrm{H}}\boldsymbol{P}_2^{\mathrm{H}}=\boldsymbol{P}_2\boldsymbol{A}_2\boldsymbol{P}_2^{\mathrm{H}}=\begin{pmatrix}\lambda_1 & \boldsymbol{b}_1^{\mathrm{T}}\hat{\boldsymbol{P}}_2^{\mathrm{H}} & \boldsymbol{0}\\ \boldsymbol{0} & \lambda_2 & \boldsymbol{b}_2^{\mathrm{T}}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{B}_3\end{pmatrix},$$

因此，至多经 $n-1$ 次收缩就把 $\boldsymbol{A}$ 约化为上三角阵，其对角元为 $\boldsymbol{A}$ 的全部特征值.

现给出 $\boldsymbol{P}_1$ 和 $\boldsymbol{A}_2$ 的运算公式：

$$\begin{cases}k_1=-\operatorname{sgn}(\boldsymbol{e}_1^{\mathrm{T}}\boldsymbol{x}_1)\|\boldsymbol{x}_1\|_2,\\ \boldsymbol{u}=\boldsymbol{x}-k_1\boldsymbol{e}_1,\\ \|\boldsymbol{u}\|_2^2=2\|\boldsymbol{x}_1\|_2(\|\boldsymbol{x}_1\|_2+|\boldsymbol{e}_1^{\mathrm{T}}\boldsymbol{x}_1|),\\ \boldsymbol{P}_1^{\mathrm{H}}=\boldsymbol{P}_1=\boldsymbol{I}-\dfrac{2\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}}{\|\boldsymbol{u}\|_2^2},\end{cases}$$

计算 $\boldsymbol{A}_2=\boldsymbol{P}_1\boldsymbol{A}\boldsymbol{P}_1^{\mathrm{H}}$ 的三个矩阵相乘等价于

$$\begin{cases}\beta=\dfrac{2}{\|\boldsymbol{u}\|_2^2},\boldsymbol{h}=\beta\boldsymbol{A}\boldsymbol{u},\boldsymbol{q}^{\mathrm{H}}=\beta\boldsymbol{u}^{\mathrm{H}}\boldsymbol{A},\gamma=\beta\boldsymbol{u}^{\mathrm{H}}\boldsymbol{h},\\ \boldsymbol{A}_2=\boldsymbol{A}-\boldsymbol{h}\boldsymbol{u}^{\mathrm{H}}-\boldsymbol{u}\boldsymbol{q}^{\mathrm{H}}+\gamma\boldsymbol{u}\boldsymbol{u}^{\mathrm{H}}.\end{cases}$$

若 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$为对称阵，则 $\boldsymbol{u}$ 可取实向量. 因此 $\boldsymbol{h}=\boldsymbol{q}$，这时

$$\boldsymbol{A}_2=\boldsymbol{A}-(\boldsymbol{u}\boldsymbol{v}^{\mathrm{T}}+\boldsymbol{v}\boldsymbol{u}^{\mathrm{T}}),$$

其中$\boldsymbol{v}=\boldsymbol{h}-\frac{\gamma}{2}\boldsymbol{u}$. 因 $\boldsymbol{A}$ 对称，由式(8.3.10)可知，$\boldsymbol{b}_1^{\mathrm{T}}=\boldsymbol{0}^{\mathrm{T}}$ 且 $\boldsymbol{B}_2\in\mathbb{R}^{(n-1)\times(n-1)}$也对称，最后将 $\boldsymbol{A}$ 约化为对角阵.

当 $\boldsymbol{A}$ 为大型稀疏阵时，这种收缩方法的不足是收缩过程不能保持 $\boldsymbol{A}$ 的稀疏性.

8.3.4 反幂法和原点位移

设 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 非奇异且可对角化，它的特征值满足$|\lambda_1|\geqslant|\lambda_2|\geqslant\cdots\geqslant|\lambda_{n-1}|>|\lambda_n|$，则$\boldsymbol{A}^{-1}$的特征值满足

$$|\lambda_n^{-1}|>|\lambda_{n-1}^{-1}|\geqslant\cdots\geqslant|\lambda_1^{-1}|,$$

即 λ_n^{-1} 为 $\boldsymbol{A}^{-1}$的主特征值，$\boldsymbol{x}_n$ 是相应的特征向量.

将幂法 8.3.1 用于 $\boldsymbol{A}^{-1}$，就是**反幂法**(inverse power method)，可以用来求 $\boldsymbol{A}$ 的按模最小的特征值及相应的特征向量. 将幂法中的 $\boldsymbol{y}^{(k)}=\boldsymbol{A}^{-1}\boldsymbol{x}^{(k-1)}$ 改为 $\boldsymbol{A}\boldsymbol{y}^{(k)}=\boldsymbol{x}^{(k-1)}$，在迭代开始前，对 $\boldsymbol{A}$ 作 LU 三角分解.

算法 8.3.9　反幂法

(1) 任给 $\boldsymbol{x}^{(0)}\in\mathbb{R}^n$，给定误差界 $\varepsilon>0$

(2) 对 $\boldsymbol{A}$ 作 LU 分解使 $\boldsymbol{A}=\boldsymbol{L}\boldsymbol{U}$(其中 $\boldsymbol{L}$ 为单位下三角阵，$\boldsymbol{U}$ 为上三角阵)

(3) 对于 $k=1,2,\cdots$

(4) 　求解 $\boldsymbol{L}\boldsymbol{U}\boldsymbol{y}^{(k)}=\boldsymbol{x}^{(k-1)}$ 即分别求解三角形方程组 $\boldsymbol{L}\boldsymbol{z}=\boldsymbol{x}^{(k-1)}$ 和 $\boldsymbol{U}\boldsymbol{y}^{(k)}=\boldsymbol{z}$

(5) 　$m_k=\max(\boldsymbol{y}^{(k)})$

(6) 　如果$|m_k-m_{k-1}|/|m_k|<\varepsilon$，则终止，否则

(7) 　$\boldsymbol{x}^{(k)}=\boldsymbol{y}^{(k)}/m_k$

(8) 　k 尾

上述算法中，$\max(\boldsymbol{y}^{(k)})$表示 $\boldsymbol{y}^{(k)}$ 的按模最大的分量. 对任意

$\boldsymbol{x}^{(0)} \in \mathbb{R}^n$，有 $\boldsymbol{x}^{(0)} = \sum_{i=1}^{n} \beta_i \boldsymbol{x}_i$，只要 $\beta_n \neq 0$，则反幂法生成的迭代序列具有收敛性质：

$$\lim_{k\to\infty} \boldsymbol{x}^{(k)} = \frac{\boldsymbol{x}_n}{\max(\boldsymbol{x}_n)},$$

$$\lim_{k\to\infty} m_k = \lim_{k\to\infty} \frac{1}{\lambda_n}\left[1 + O\left(\left|\frac{\lambda_n}{\lambda_{n-1}}\right|^k\right)\right] = \frac{1}{\lambda_n},$$

且迭代收敛速度决定于 $\left|\frac{\lambda_n}{\lambda_{n-1}}\right|$.

设非零常数 $s \notin \sigma(\boldsymbol{A})$，即 $s \neq \lambda_i, i=1,2,\cdots,n$，称 $\boldsymbol{A}-s\boldsymbol{I}$ 为矩阵 $\boldsymbol{A}$ 的原点位移，s 称为**位移**（shift），$(\boldsymbol{A}-s\boldsymbol{I})^{-1}$ 的特征值为 $(\lambda_i - s)^{-1}, i=1,\cdots,n$，而特征向量不变.

如果已知 s 接近于 $\boldsymbol{A}$ 的某个特征值 λ_j，即 $s \approx \lambda_j$，且

$$0 < |\lambda_j - s| \ll |\lambda_j - \lambda_i|, \quad j \neq i, \quad i = 1,2,\cdots,n, \tag{8.3.12}$$

对 $\boldsymbol{A}-s\boldsymbol{I}$ 用反幂法 8.3.9，仅将第 2 步中的 $\boldsymbol{A}$，用 $\boldsymbol{A}-s\boldsymbol{I}$ 代替，就是原点位移反幂法，此方法有收敛性质：

$$\lim_{k\to\infty} m_k = \frac{1}{\lambda_j - s}, \quad \lim_{k\to\infty} \boldsymbol{x}^{(k)} = \frac{\boldsymbol{x}_j}{\max(\boldsymbol{x}_j)},$$

可求得 λ_j 和相应的 $\boldsymbol{x}_j$. 其收敛速度取决于比值

$$r = \max_{\lambda_i \neq \lambda_j} \frac{|\lambda_j - s|}{|\lambda_i - s|}.$$

因此，只要 s 是 λ_j 的一个较好的近似且满足式(8.3.12)，收敛是快的. 原点位移反幂法通常是求解单个特征值和特征向量的有效的方法.

8.4 QR 算法

QR 算法是求矩阵特征值的最有效和应用最广泛的一种方法.

8.4.1 基本算法及收敛性

QR 算法也是一种变换方法. 设 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$，令 $\boldsymbol{A}_1=\boldsymbol{A}$. 先将 $\boldsymbol{A}_1$ 作 QR 分解（见 8.2.4 节），写成 $\boldsymbol{A}_1=\boldsymbol{QR}$，其中 $\boldsymbol{Q}$ 是酉矩阵即 $\boldsymbol{Q}^{\mathrm{H}}\boldsymbol{Q}=\boldsymbol{I}$，$\boldsymbol{Q}^{\mathrm{H}}=\overline{\boldsymbol{Q}}^{\mathrm{T}}$ 是 $\boldsymbol{Q}$ 的共轭转置，$\boldsymbol{R}$ 是上三角阵，当 $\boldsymbol{A}$ 非奇异且规定 $\boldsymbol{R}$ 的对角元是正实数时，则分解是惟一的，然后令 $\boldsymbol{A}_2=\boldsymbol{RQ}$，则有 $\boldsymbol{A}_2=\boldsymbol{Q}^{\mathrm{H}}\boldsymbol{A}_1\boldsymbol{Q}$. $\boldsymbol{A}_2$ 是 $\boldsymbol{A}_1$ 的正交相似变换，它们有相同的特征值. 这个过程可继续下去，得到迭代序列 $\{\boldsymbol{A}_k\}$，称此过程为基本 QR 算法过程.

算法 8.4.1 基本 QR 算法

(1) 定义 $\boldsymbol{A}_1=\boldsymbol{A}$

(2) 对于 $k=1,2,\cdots$

(3) $\quad\boldsymbol{A}_k=\boldsymbol{Q}_k\boldsymbol{R}_k$（对 $\boldsymbol{A}_k$ 作 QR 分解）

(4) $\quad\boldsymbol{A}_{k+1}=\boldsymbol{Q}_k^{\mathrm{H}}\boldsymbol{A}_k\boldsymbol{Q}_k=\boldsymbol{R}_k\boldsymbol{Q}_k$

定理 8.4.2 QR 算法产生的序列满足

(1) $\boldsymbol{A}_{k+1}=\boldsymbol{Q}_k^{\mathrm{H}}\boldsymbol{A}_k\boldsymbol{Q}_k$

(2) $\boldsymbol{A}_{k+1}=\hat{\boldsymbol{Q}}_k^{\mathrm{H}}\boldsymbol{A}\hat{\boldsymbol{Q}}_k$，其中 $\hat{\boldsymbol{Q}}_k=\boldsymbol{Q}_1\boldsymbol{Q}_2\cdots\boldsymbol{Q}_k$

(3) $\boldsymbol{A}^k=\hat{\boldsymbol{Q}}_k\hat{\boldsymbol{R}}_k$，其中 $\hat{\boldsymbol{R}}_k=\boldsymbol{R}_k\cdots\boldsymbol{R}_2\boldsymbol{R}_1$

定理说明：迭代序列 $\{\boldsymbol{A}_k\}$ 保持 $\boldsymbol{A}$ 的特征值不变；QR 算法是 QR 变换过程，实质是对 $\boldsymbol{A}^k$ 作 QR 分解.

定义 8.4.3 当 $k\to\infty$ 时的矩阵序列 $\{\boldsymbol{A}_k\}$，若其对角元均收敛且严格下三角部分元素收敛到零，则称 $\{\boldsymbol{A}_k\}$ **基本收敛**（basic convergence）**到上三角阵**.

以上定义未指出 $\{\boldsymbol{A}_k\}$ 的严格上三角部分元素是否收敛. 但对 $\boldsymbol{A}$ 的特征值而言，基本收敛就够了. 基本 QR 算法有如下的基本收敛定理.

定理 8.4.4 设 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$ 的特征值满足

$$|\lambda_1|>|\lambda_2|>\cdots>|\lambda_n|>0, \tag{8.4.1}$$

λ_i 对应的右特征向量为 $\boldsymbol{x}_i \in \mathbb{C}^n, i=1,2,\cdots,n$. 构成方阵 $\boldsymbol{X}=(\boldsymbol{x}_1, \boldsymbol{x}_2,\cdots,\boldsymbol{x}_n)$，设 $\boldsymbol{X}^{-1}$ 可分解为 $\boldsymbol{X}^{-1}=\boldsymbol{LU}$，其中 $\boldsymbol{L}$ 为单位下三角阵，$\boldsymbol{U}$ 为上三角阵，则基本 QR 算法产生的序列 $\{\boldsymbol{A}_k\}$ 基本收敛到上三角阵(当 $\boldsymbol{A}$ 对称时，$\{\boldsymbol{A}_k\}$ 收敛到对角阵)，其对角元极限为

$$\lim_{k\to\infty} a_{ii}^{(k)} = \lambda_i, i = 1,2,\cdots,n$$

在该定理的条件下，基本 QR 算法的收敛速度取决于 $\max\limits_{1\leqslant i\leqslant n-1}\left|\dfrac{\lambda_{i+1}}{\lambda_i}\right|$. 如果比值 $\left|\dfrac{\lambda_{i+1}}{\lambda_i}\right|$ 接近于 1，那么 $\{\boldsymbol{A}_k\}$ 的基本收敛速度将十分缓慢，因此有必要采用 8.4.3 节的原点位移 QR 算法.

如果不满足定理 8.4.4 的条件，则 QR 算法产生的序列 $\{\boldsymbol{A}_k\}$ 可能不收敛. 例如，设 $\boldsymbol{A}=\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}$，有特征值 $\lambda=\pm 1$，不满足定理 8.4.4 的条件. 对 $\boldsymbol{A}_1=\boldsymbol{A}$ 作 QR 分解，即 $\boldsymbol{A}_1=\boldsymbol{Q}_1\boldsymbol{R}_1$，其中 $\boldsymbol{Q}_1=\boldsymbol{A}$，$\boldsymbol{R}_1=\boldsymbol{I}$，所以 $\boldsymbol{A}_2=\boldsymbol{R}_1\boldsymbol{Q}_1=\boldsymbol{A}$，也就是说 $\boldsymbol{A}_k=\boldsymbol{A}$，$\{\boldsymbol{A}_k\}$ 不收敛到对角阵.

一般情况 QR 算法的收敛性较为复杂. 在有等模特征值(包括实的重特征值，单的或重的共轭复特征值及其他情况)时，如果 $\boldsymbol{X}^{-1}$ 仍有 LU 分解，则 $\{\boldsymbol{A}_k\}$ 基本收敛于块上三角阵. 具体地说，如果有 p 个按模相同的特征值，则在对角线上出现一个 p 阶子阵，其特征值收敛于 $\boldsymbol{A}$ 的等模特征值. 特别地，当等模特征值中只有实的重特征值和多重复特征值时，这种分块上三角阵的对角块是 1 阶或 2 阶的，1 阶的元素以及 2 阶子阵的一对复共轭特征值收敛于 $\boldsymbol{A}$ 的特征值. 然而分块上三角阵的对角线以上元素和 p 阶子阵(包括 2 阶子阵)的元素不一定收敛. 这就是**基本收敛**.

QR 算法适合于求解中小型特征值问题且当矩阵特征值分布非常稠密时，QR 算法可能不收敛.

8.4.2 海森伯格阵的 QR 算法

设 $A\in\mathbb{R}^{n\times n}$，当 A 为一般矩阵时，QR 算法的计算量很大. 在实际计算时为了节省计算量，总是先将 A 作正交相似变换化为上海森伯格阵，然后对上海森伯格阵施行 QR 算法，求得矩阵 A 的全部特征值.

算法 8.4.5 上海森柏格阵的 QR 算法

(1) 将 A 正交相似变换为上海森伯格阵 A_1（选用豪斯霍尔德变换阵，见 8.2.3 节）

(2) 对于 $k=1,2,\cdots$

$A_k=Q_kR_k$（选用吉文斯变换对 A_k 作 QR 分解，见算法 8.2.13）

$A_{k+1}=Q_k^{\mathrm{T}}A_kQ_k=R_kQ_k$

算法中的 Q_k 和 A_k 都是上海森伯格形. 并假设 A_k 是不可约的. 假如 A_k 的一个或几个次对角元非常接近于零，那么可把 A_k 划分为较小型的上海森伯格阵块. 阶数小于等于 2 的对角块的特征值就是 A 的特征值，称这种策略为划分和收缩.

8.4.3 原点位移的 QR 算法

基本 QR 算法（包括上海森伯格阵的 QR 算法）的收敛速度与幂法一样，依赖于相邻特征值的比值，因而一般是线性的. 为了加速收敛，类似于反幂法的原点位移（见 8.3.4 节），引进原点位移的 QR 算法.

算法 8.4.6 原点位移 QR 算法

(1) 将 $A\in\mathbb{R}^{n\times n}$ 正交相似变换为上海森伯格阵 A_1（选用豪斯霍尔德变换阵，见 8.2.3 节）

(2) 对于 $k=1,2,\cdots$

① 选位移 μ_k

② $A_k - \mu_k I = Q_k R_k$（选用吉文斯变换对 $A_k - \mu_k I$ 作 QR 分解，见 8.2.4 节）

③ $A_{k+1} = Q_k^T A_k Q_k = R_k Q_k + \mu_k I$

算法生成的矩阵序列$\{A_k\}$和$\{Q_k\}$都是上海森伯格阵且 A_k 与 A 相似. 选位移 μ_k 的原则是：使算法的收敛速度加快. 为方便，记 μ_k 为 μ，选 μ 使 $A - \mu I$ 的特征值有如下次序：

$$|\lambda_1 - \mu| \geqslant \cdots \geqslant |\lambda_n - \mu|,$$

则 A_k 的第 j 个次对角元收敛速度决定于 $\left|\dfrac{\lambda_{j+1} - \mu}{\lambda_j - \mu}\right|^k$. 如果 μ 十分靠近 λ_n，则收敛会很快.

矩阵 A 满足定理 8.4.4 的条件下，由算法 8.4.1 生成的 A_k 的右下角 2 阶子阵

$$\begin{pmatrix} a_{n-1,n-1}^{(k)} & a_{n-1,n}^{(k)} \\ a_{n,n-1}^{(k)} & a_{n,n}^{(k)} \end{pmatrix}$$

的特征值 $\lambda_{n-1}^{(k)}$ 和 $\lambda_n^{(k)}$，则

$$\lim_{k\to\infty} a_{nn}^{(k)} = \lambda_n, \quad \lim_{k\to\infty} \lambda_{n-1}^{(k)} = \lambda_{n-1}, \quad \lim_{k\to\infty} \lambda_n^{(k)} = \lambda_n.$$

因此，选 μ_k 与 A 的特征值很靠近的值. 一个合理的选择是 $\mu_k = a_{nn}^{(k)}$，称为**瑞利商位移**. 然而一个更有效的选择是：当迭代过程中

$$\begin{pmatrix} a_{n-1,n-1}^{(k)} & a_{n-1,n}^{(k)} \\ a_{n,n-1}^{(k)} & a_{n,n}^{(k)} \end{pmatrix} \tag{8.4.2}$$

的两个特征值是实数时，选 $\lambda_{n-1}^{(k)}$ 和 $\lambda_n^{(k)}$ 中最接近于 $a_{nn}^{(k)}$ 的那个作为 μ_k，称为**维尔金森(Wilkinson)位移**.

在一定条件下，原点位移 QR 算法产生的$\{A_k\}$基本收敛到一个上三角阵，其对角元为矩阵 A 的特征值且收敛速度比基本 QR 算法快，基本 QR 算法只是线性收敛. 若 $a_{n,n-1}^{(k)}$ 收敛于零，一定最先趋于零，最先得到最小的一个特征值 λ_n，则其收敛速度可能是二次的，即收敛速度取决于 $\max\limits_{1\leqslant i\leqslant n-1}\left|\dfrac{\lambda_{i+1} - \mu}{\lambda_i - \mu}\right|^2$.

8.4.4 双步隐式 QR 算法

如果 $\boldsymbol{A}$ 有复特征值，则 QR 算法产生的 $\boldsymbol{A}_k$ 其元素 $a_{nn}^{(k)}$ 是一个很坏的近似特征值. 这时算法 8.4.6 即便收敛，也是极其慢的，为此引入双步隐式 QR 算法，其核心是双步隐式 QR 变换.

设 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 是不可约上海森伯格阵，它的右下角 2 阶子阵

$$\begin{pmatrix} a_{n-1,n-1} & a_{n-1,n} \\ a_{n,n-1} & a_{nn} \end{pmatrix}$$

的特征值 μ_1 和 μ_2，它们可能都是实数，也可能是一对共轭复数，记

$$s = a_{n-1,n-1} + a_{nn} = \mu_1 + \mu_2 \in \mathbb{R},$$

$$t = a_{n-1,n-1}a_{nn} - a_{n,n-1}a_{n-1,n} = \mu_1\mu_2 \in \mathbb{R},$$

取 μ_1 和 μ_2 为位移，在复域上连续作两次单步原点位移 QR 变换即双步 QR 变换：

$$\begin{cases} \boldsymbol{A} - \mu_1\boldsymbol{I} = \boldsymbol{Q}_1\boldsymbol{R}_1, & \text{（复 QR 分解）} \\ \boldsymbol{B} = \boldsymbol{R}_1\boldsymbol{Q}_1 + \mu_1\boldsymbol{I}, & \\ \boldsymbol{B} - \mu_2\boldsymbol{I} = \boldsymbol{Q}_2\boldsymbol{R}_2, & \text{（复 QR 分解）} \\ \boldsymbol{C} = \boldsymbol{R}_2\boldsymbol{Q}_2 + \mu_2\boldsymbol{I}, & \end{cases} \tag{8.4.3}$$

其中 $\boldsymbol{A}-\mu_1\boldsymbol{I}$ 和 $\boldsymbol{B}-\mu_2\boldsymbol{I}$ 分别在复数域上作 QR 分解，$\boldsymbol{Q}_1$ 和 $\boldsymbol{Q}_2$ 为酉阵，$\boldsymbol{R}_1$ 和 $\boldsymbol{R}_2$ 为上三角阵. 令

$$\boldsymbol{M} = (\boldsymbol{A}-\mu_2\boldsymbol{I})(\boldsymbol{A}-\mu_1\boldsymbol{I}) = \boldsymbol{A}^2 - s\boldsymbol{A} + t\boldsymbol{I} \in \mathbb{R}^{n\times n}$$

为实阵. 则上述复数域上双步 QR 变换 (8.4.3)等价于下面的实矩阵 $\boldsymbol{M}$ 的一步实 QR 变换：

$$\begin{cases} \boldsymbol{M} = \boldsymbol{A}^2 - s\boldsymbol{A} + t\boldsymbol{I}, & \\ \boldsymbol{M} = \boldsymbol{Q}\boldsymbol{R}, & \text{（实 QR 分解）} \\ \boldsymbol{C} = \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}, & \end{cases} \tag{8.4.4}$$

其中 $\boldsymbol{M}$ 是在实域上作 QR 分解.

易验证变换(8.4.4)与变换(8.4.3)的等价关系.

$$\begin{aligned}\boldsymbol{M} &= (\boldsymbol{A}-\mu_2\boldsymbol{I})(\boldsymbol{A}-\mu_1\boldsymbol{I}) = (\boldsymbol{A}-\mu_2\boldsymbol{I})\boldsymbol{Q}_1\boldsymbol{R}_1 = \boldsymbol{Q}_1(\boldsymbol{Q}_1^{\mathrm{H}}\boldsymbol{A}\boldsymbol{Q}_1-\mu_2\boldsymbol{I})\boldsymbol{R}_1 \\ &= \boldsymbol{Q}_1(\boldsymbol{B}-\mu_2\boldsymbol{I})\boldsymbol{R}_1 = \boldsymbol{Q}_1\boldsymbol{Q}_2\boldsymbol{R}_2\boldsymbol{R}_1 = \boldsymbol{Q}\boldsymbol{R},\end{aligned}$$

其中 $\boldsymbol{Q}=\boldsymbol{Q}_1\boldsymbol{Q}_2$, $\boldsymbol{R}=\boldsymbol{R}_2\boldsymbol{R}_1$. 可以选取 $\boldsymbol{Q}_1$ 和 $\boldsymbol{Q}_2$, 使 $\boldsymbol{Q}$ 为实的正交阵. 由变换(8.4.3),得

$$\boldsymbol{C} = \boldsymbol{Q}_2^{\mathrm{H}}\boldsymbol{B}\boldsymbol{Q}_2 = \boldsymbol{Q}_2^{\mathrm{H}}\boldsymbol{Q}_1^{\mathrm{H}}\boldsymbol{A}\boldsymbol{Q}_1\boldsymbol{Q}_2 = \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}.$$

变换(8.4.4)的第一步计算量很大,要尽量减少运算量,并用下述算法实现.

算法 8.4.7 双步隐式 QR 变换

(1) 定义 $\boldsymbol{A}$ 为不可约上海森伯格阵

(2) 计算 $\boldsymbol{M}$ 的第 1 列 $\boldsymbol{M}\boldsymbol{e}_1=(m_{11},m_{21},m_{31},0,\cdots,0)^{\mathrm{T}}$

(3) 确定豪斯霍尔德变换阵 $\boldsymbol{P}_0=\boldsymbol{I}-2\dfrac{\boldsymbol{u}_0\boldsymbol{u}_0^{\mathrm{T}}}{\|\boldsymbol{u}_0\|_2^2}$, 使 $\boldsymbol{P}_0(\boldsymbol{M}\boldsymbol{e}_1)$ 是 $\boldsymbol{e}_1$ 的倍数.

$$\boldsymbol{P}_0 = \begin{pmatrix}\hat{\boldsymbol{P}}_0 & \\ & \boldsymbol{I}_{n-3}\end{pmatrix}, \quad \text{其中} \hat{\boldsymbol{P}}_0 \in \mathbb{R}^{3\times 3}$$

(4) 对 $\boldsymbol{A}$ 作正交相似变换,即形如

$$\boldsymbol{P}_0\boldsymbol{A}\boldsymbol{P}_0 = \begin{pmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ \circledast & * & * & * & * & * \\ \circledast & \circledast & * & * & * & * \\ & & & * & * & * \\ & & & & * & * \end{pmatrix},$$

其中⊛为新增非零元素.

(5) 计算豪斯霍尔德变换阵 $\boldsymbol{P}_1,\cdots,\boldsymbol{P}_{n-2}$, 将 $\boldsymbol{P}_0\boldsymbol{A}\boldsymbol{P}_0$ 逐次恢复为上海森伯格阵,使得

$$P_{n-2}\cdots P_1 P_0 A P_0 P_1 \cdots P_{n-2} = Q^{\mathrm{T}} A Q = D,$$

其中 $Q=P_0P_1\cdots P_{n-2}$.

$$P_i = \begin{pmatrix} I_i & & \\ & \hat{P}_i & \\ & & I_{n-i-3} \end{pmatrix}, \quad \hat{P}_i \in \mathbb{R}^{3\times 3}, \quad i=1,\cdots,n-3$$

$$P_{n-2} = \begin{pmatrix} I_{n-2} & \\ & \hat{P}_{n-2} \end{pmatrix}, \quad \hat{P}_{n-2} \in \mathbb{R}^{2\times 2}$$

以上算法中的 D 在本质上就是实 QR 变换(8.4.4)中的矩阵 C.

下面的双步隐式 QR 算法是计算 $A\in\mathbb{R}^{n\times n}$ 的全部特征值(含复特征值)的迭代法. 在每次迭代中都需要执行双步隐式 QR 变换算法 8.4.7,且将其具体化了. 其中 $A(1:3,1:n)$ 表示矩阵 A 的前 3 行; $A(j+1:k,j)$ 表示矩阵 A 的第 j 列的第 $j+1$ 到 k 的元素等.

算法 8.4.8 双步隐式 QR 算法

(1) 将 A 约化为上海森伯格阵且不可约

(2) 对于 $k=1,2,\cdots$

① 计算参数

$$s = a_{n-1,n-1} + a_{nn}$$
$$t = a_{n-1,n-1}a_{nn} - a_{n,n-1}a_{n-1,n}$$
$$m_{11} = h_{11}^2 + h_{12}h_{21} - sh_{11} + t$$
$$m_{21} = h_{21}(h_{11} + h_{22} - s)$$
$$m_{31} = h_{21}h_{32}$$

② 对矩阵 A 作初始正交相似变换:

记 $x=(m_{11},m_{21},m_{31})^{\mathrm{T}}$,构造豪斯霍尔德变换阵 P,使$Px=\sigma e_1$.

令 $i=\min(n,4)$,计算

$\boldsymbol{A}(1:3,1:n):=\boldsymbol{P}*\boldsymbol{A}(1:3,1:n)$

$\boldsymbol{A}(1:i,1:3):=\boldsymbol{A}(1:i,1:3)*\boldsymbol{P}$

③ 化矩阵 **A** 为上海森伯格形

对于 $j=1,2,\cdots,n-2$

令 $i=\min(j+3,n),q=\min(i+1,n)$

记 $\boldsymbol{x}=\boldsymbol{A}(j+1:i,j)$，构造豪斯霍尔德变换阵 **P**，使 $\boldsymbol{Px}=\sigma\boldsymbol{e}_1$.

$\boldsymbol{A}(j+1:i,j:n):=\boldsymbol{P}*\boldsymbol{A}(j+1:i,j:n)$

$\boldsymbol{A}(1:q,j+1:i):=\boldsymbol{A}(1:q,j+1:i)*\boldsymbol{P}$

其中的变换均在原矩阵 **A** 上进行，即不保留原矩阵. 实际计算时尚需作收敛性判断. 首先判断上海森伯格阵 **A** 是否可约. 对于给定的绝对误差容限 $\varepsilon>0$（如 $\varepsilon=10^{-5}$），从下到上判断次对角元是否满足

$$|a_{k+1,k}|<\varepsilon,\quad k=n-1,\cdots,2,1,\tag{8.4.5}$$

若满足，则视 $a_{k+1,k}$ 为零，对 **A** 进行划分和收缩. 然后判断对角块的阶数. 对于阶数小于等于 2 的对角块，求出它们的特征值. 对于阶数大于 3 的对角块，重复以上迭代和判断过程，直到求出原矩阵 **A** 的全部特征值.

例 8.4.9 设矩阵

$$\boldsymbol{A}=\begin{pmatrix}3&2&3&4&5&6&7\\11&1&2&3&4&5&6\\2&8&9&1&2&3&4\\-4&2&9&11&13&15&8\\-1&-2&-3&-1&-1&-1&-1\\3&2&3&4&13&15&8\\-2&-2&-3&-4&-5&-3&-3\end{pmatrix}.$$

已知其特征值为 18.4123，11.1805，1.7099±4.2522i，4.4983，−2.2327，−0.2783. 试用划分和收缩技术，分别用算法 8.4.5 和算法 8.4.8 求 **A** 的全部特征值，取误差容限 $\varepsilon=5\times10^{-5}$.

解 为简单起见，在使用划分和收缩技术时，每次只用条件(8.4.5)划分出右下角的一个1阶或2阶对角块，并对剩下的左上角对角块(不管它是否可约)继续迭代和划分.

(1) 算法8.4.5的计算结果如下：

对于$\boldsymbol{A}$的海森伯格形迭代到第4次，划分出1阶对角块，$\lambda=-0.2783$.

对于剩下的6阶对角块迭代到第19次，划分出1阶对角块，$\lambda=-2.2327$.

对于剩下的5阶对角块迭代到第536次，划分出1阶对角块，$\lambda=4.4982$.

对于剩下的4阶对角块迭代到第537次，划分出2阶对角块

$$\begin{pmatrix} 2.6162 & -4.9453 \\ 3.8223 & 0.8037 \end{pmatrix},$$

它的特征值是$\lambda=1.7100\pm4.2522\mathrm{i}$，$|\lambda|=4.5831$.

最后剩下的2阶对角块是

$$\begin{pmatrix} 18.4123 & -13.3649 \\ 0.0000 & 11.1805 \end{pmatrix},$$

它的特征值是$\lambda=11.1805, 18.4123$.

可见当$\boldsymbol{A}$有复特征值时，算法8.4.5的收敛速度很慢，此例还证实了，在一定条件下基本QR算法收敛到舒尔分块上三角形，并且对角块按特征值的模从大到小排列.

(2) 双步隐式QR算法8.4.8的计算结果如下：

对于$\boldsymbol{A}$的海森伯格形迭代到第3次，划分出1阶对角块，$\lambda=-0.2783$.

对于剩下的6阶对角块迭代到第5次，划分出1阶对角块，$\lambda=4.4983$.

对于剩下的5阶对角块迭代到第6次，划分出1阶对角块，$\lambda=-2.2327$.

对于剩下的 4 阶对角块迭代到第 8 次，划分出 2 阶对角块

$$\begin{pmatrix} 0.9086 & -3.6806 \\ 5.0870 & 2.5112 \end{pmatrix},$$

它的特征值是 $\lambda=1.7099\pm4.2522\mathrm{i}$.

最后剩下的 2 阶对角块是

$$\begin{pmatrix} 18.4113 & -13.3655 \\ -0.0006 & 11.1816 \end{pmatrix},$$

它的特征值是 $\lambda=18.4123, 11.1805$.

8.5 对称 QR 算法

设 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 是对称阵，用带位移的 QR 算法求解对称特征值问题，其实施过程可大大简化. (1)用豪斯霍尔德正交相似变换 $\boldsymbol{A}$ 为海森伯格阵时，得到的是一个对称三对角阵 $\boldsymbol{T}$. (2)进行每步带位移的 QR 迭代时，其结果矩阵仍保持三对角形. (3)因为实对称矩阵 $\boldsymbol{A}$ 的特征值是实的，因而只需用单原点位移的 QR 算法，而不必考虑在有复特征值情况下用的双步隐式 QR 算法.

下面考虑实对称矩阵的隐位移 QR 算法. 首先用豪斯霍尔德变换将 $\boldsymbol{A}$ 正交相似变换为一个对称三对角阵 $\boldsymbol{T}$，并简化计算. 设已找到豪斯霍尔德阵 $\boldsymbol{P}_1,\cdots,\boldsymbol{P}_{k-1}$，使

$$\boldsymbol{A}_{k-1}=(\boldsymbol{P}_1\cdots\boldsymbol{P}_{k-1})^{\mathrm{T}}\boldsymbol{A}(\boldsymbol{P}_1\cdots\boldsymbol{P}_{k-1})$$

$$=\begin{pmatrix} \multicolumn{2}{c}{\boldsymbol{B}_k} & \begin{matrix}\boldsymbol{0}\\ \boldsymbol{b}^{\mathrm{T}}\end{matrix} \\ \boldsymbol{0} & \boldsymbol{b} & \boldsymbol{C} \end{pmatrix}\begin{matrix} k-1 \\ 1 \\ n-k \end{matrix},$$

$$\quad\; k-1 \quad 1 \qquad n-k$$

其中 $\boldsymbol{B}_k$ 是 $k\times k$ 的三对角阵. 如果找到 $n-k$ 阶的豪斯霍尔德阵 $\bar{\boldsymbol{P}}_k$，使 $\bar{\boldsymbol{P}}_k\boldsymbol{b}$ 为 $\boldsymbol{e}_1\in\mathbb{R}^{n-k}$ 的倍数，则取 $\boldsymbol{P}_k=\mathrm{diag}(\boldsymbol{I}_k,\bar{\boldsymbol{P}}_k)\in\mathbb{R}^{n\times n}$，有

$$A_k = P_k A_{k-1} P_k = \left[\begin{array}{c|c} \begin{matrix} B_k \end{matrix} & \begin{matrix} 0 \\ \hline b^{\mathrm{T}} \bar{P}_k \end{matrix} \\ \hline \begin{matrix} 0 & \bar{P}_k b \end{matrix} & \bar{P}_k C \bar{P}_k \end{array}\right].$$

将 A_k 重新划分出 $(k+1)\times(k+1)$ 的三对角子阵 B_{k+1}. 继续进行，直到 $n-2$ 步. 令 $Q = P_1 P_2 \cdots P_{n-2}$，则 $Q^{\mathrm{T}} A Q = T$ 为三对角阵.

在 A_k 的计算过程中，$\bar{P}_k C \bar{P}_k$ 的计算充分利用对称性. 其中

$$\bar{P}_k = I - \beta u u^{\mathrm{T}}, u \in \mathbb{R}^{n-k}, u \neq 0, \beta = 2/\|u\|_2^2.$$

令

$$z = \beta C u, \quad w = z - \frac{\beta}{2}(z^{\mathrm{T}} u) u,$$

则

$$\bar{P}_k C \bar{P}_k = C - u w^{\mathrm{T}} - w u^{\mathrm{T}}.$$

由于只需计算此矩阵的上三角部分，这样，从 A_{k-1} 到 A_k 的变换只需 $4(n-k)^2$ 个 flop.

算法 8.5.1　豪斯霍尔德三对角化

(1) 给定一对称阵 $A \in \mathbb{R}^{n\times n}$，本算法计算 $T = Q^{\mathrm{T}} A Q$ 并覆盖 A，其中 T 为三对角阵，$Q = P_1 \cdots P_{n-2}$ 是豪斯霍尔德变换阵的乘积.

(2) 对于 $k = 1, 2, \cdots, n-2$

$u = (A(k+1,k) + \mathrm{sgn}(A(k+1,k)))\|A(k+1:n,k)\|_2,$
　　$A(k+2,k), \cdots, A(n,k))^{\mathrm{T}}$

$\beta = 2/\|u\|_2^2$

$z = \beta A(k+1:n, k+1:n) u$

$w = z - (\beta z^{\mathrm{T}} u/2) u$

$A(k+1,k) = \|A(k+1:n,k)\|_2$

$A(k,k+1) = A(k+1,k)$

$A(k+1:n, k+1:n) = A(k+1:n, k+1:n) - u w^{\mathrm{T}} - w u^{\mathrm{T}}$

k 尾

下面对三对角阵 $\boldsymbol{T}$ 进行带位移的 QR 迭代：

$$\begin{cases} \boldsymbol{T}_k - \mu_k \boldsymbol{I} = \boldsymbol{Q}_k \boldsymbol{R}_k \quad (\text{QR 分解}), \\ \boldsymbol{T}_{k+1} = \boldsymbol{Q}_k^{\mathrm{T}} \boldsymbol{T}_k \boldsymbol{Q}_k = \boldsymbol{R}_k \boldsymbol{Q}_k + \mu_k \boldsymbol{I}, \end{cases}$$

其中 $\boldsymbol{T}_k$ 是一个对称三对角阵，

$$\boldsymbol{T}_k = \begin{pmatrix} a_1 & b_1 & & & \\ b_1 & a_2 & b_2 & & \\ & \ddots & \ddots & \ddots & \\ & & b_{n-2} & a_{n-1} & b_{n-1} \\ & & & b_{n-1} & a_n \end{pmatrix},$$

则位移的一个合理的选择是 $\mu_k = a_n$，称为**瑞利商位移**. 一个更有效的选择是取二阶矩阵

$$\begin{pmatrix} a_{n-1} & b_{n-1} \\ b_{n-1} & a_n \end{pmatrix}$$

的两个特征值中最接近 a_n 的特征值作为位移，称为**威尔金森位移**，即

$$\mu_k = a_n + d - \operatorname{sgn}(d)\sqrt{d^2 + b_{n-1}^2},$$

其中 $d = (a_{n-1} - a_n)/2$. 这两种位移都使 QR 迭代达到三次收敛，即其收敛速度取决于 $\max\limits_{1 \leqslant i \leqslant n-1} \left| \dfrac{\lambda_{i+1} - \mu_R}{\lambda_i - \mu_R} \right|^3$.

不必显式地形成矩阵 $\boldsymbol{T}_k - \mu_k \boldsymbol{I}$，就能实现从 $\boldsymbol{T}_k$ 到 $\boldsymbol{T}_{k+1} = \boldsymbol{R}_k \boldsymbol{Q}_k + \mu_k \boldsymbol{I} = \boldsymbol{Q}_k^{\mathrm{T}} \boldsymbol{T}_k \boldsymbol{Q}_k$ 的变换，即用隐式位移. 由于 $\boldsymbol{T} - \mu \boldsymbol{I} = \boldsymbol{QR}$，于是 $\boldsymbol{Q}$ 的第 1 列由

$$(\boldsymbol{T} - \mu \boldsymbol{I})\boldsymbol{e}_1 = (a_1 - \mu, b_1, 0, \cdots, 0)^{\mathrm{T}}$$

确定. 若令 $c = \cos\theta, s = \sin\theta$ 使得

$$\begin{pmatrix} c & s \\ -s & c \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} a_1 - \mu \\ b_1 \end{pmatrix} = \begin{pmatrix} x \\ 0 \end{pmatrix},$$

则对吉文斯变换阵 $\boldsymbol{J}_1 = \boldsymbol{J}(1, 2, \theta) \in \mathbb{R}^{n \times n}$ 成立，$\boldsymbol{J}_1 \boldsymbol{e}_1 = \boldsymbol{Q}\boldsymbol{e}_1$ 和 $\boldsymbol{J}_1^{\mathrm{T}} \boldsymbol{T} \boldsymbol{J}_1$ 有形状（以 $n=5$ 为例）

$$J_1^{\mathrm{T}}TJ_1=\begin{bmatrix} * & * & \circledast & & \\ * & * & * & & \\ \circledast & * & * & * & \\ & & * & * & * \\ & & & * & * \end{bmatrix},$$

在 $J_1^{\mathrm{T}}TJ_1$ 中产生两个非零元⊛在三对角形区域的外面，然后再对 $J_1^{\mathrm{T}}TJ_1$ 应用一系列吉文斯变换 $J_2,\cdots,J_{n-1}$，逐次消去所产生的位于三对角形区域外的元素⊛，使

$$\tilde{T}=J_{n-1}^{\mathrm{T}}\cdots J_2^{\mathrm{T}}(J_1^{\mathrm{T}}TJ_1)J_2\cdots J_{n-1}$$

是对称三对角阵. 由于 $J_1\cdots J_{n-1}$ 的第 1 列与 J_1 的第 1 列是一样的. 因此 $\tilde{T}$ 和用显式位移 QR 法获得的对角阵本质上是一样的. 从而有如下的从 T_k 到 T_{k+1} 的隐式对称 QR 步算法.

算法 8.5.2 带威尔金森位移的隐式对称 QR 步

(1) 给定一个不可约对称三对角阵 $T\in\mathbb{R}^{n\times n}$，$T=(t_{ij})_n$

(2) 计算威尔金森位移 μ 和 $T-\mu I$ 的第 1 列的前两个非零元素.

$d=(t_{n-1,n-1}-t_{nn})/2$

$\mu=t_{nn}-t_{n,n-1}^2/(d+\mathrm{sgn}(d)\sqrt{d^2+t_{n,n-1}^2})$

$x=t_{11}-\mu$

$z=t_{21}$

(3) 对于 $k=1,2,\cdots,n-1$

计算 $c=\dfrac{x}{\sqrt{x^2+z^2}}$，$s=\dfrac{-z}{\sqrt{x^2+z^2}}$ 得 $J_k=J(k,k+1,\theta)$

$T=J_k^{\mathrm{T}}TJ_k$，

如果 $k<n-1$，则

$x=t_{k+1,k}$

$z=t_{k+2,k}$

k 尾

该算法是如下的对称 QR 算法的基础.

算法 8.5.3 对称 QR 算法

给定对称阵 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$和一个比舍入误差单位大的容差 tol. 本算法计算一个近似对称舒尔分解 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}=\boldsymbol{D}$.

(1) 用算法 8.5.1 计算 $\boldsymbol{A}$ 的对称三对角化矩阵 $\boldsymbol{T}$, 即

$$\boldsymbol{T}=(\boldsymbol{P}_1\cdots\boldsymbol{P}_{n-2})^{\mathrm{T}}\boldsymbol{A}(\boldsymbol{P}_1\cdots\boldsymbol{P}_{n-2}),$$

其中 $\boldsymbol{P}_i(i=1,\cdots,n-2)$是豪斯霍尔德变换阵.

$\boldsymbol{D}=\boldsymbol{T}$.

(2) 重复进行如下运算:

① 对于 $i=1,2,\cdots,n-1$

当 $|d_{i+1,i}|=|d_{i,i+1}|\leqslant \mathrm{tol}(|d_{ii}|+|d_{i+1,i+1}|)$时,

$d_{i,i+1}=0, d_{i+1,i}=0.$

② 找最大的 q 和最小的 p, 使 $\boldsymbol{D}$ 有分块形式

$$\boldsymbol{D}=\begin{bmatrix}\boldsymbol{D}_{11} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{D}_{22} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{D}_{33}\end{bmatrix}\begin{matrix}p\\ n-p-q,\\ q\end{matrix}$$

$$\quad p \quad n-p-q \quad q$$

其中 $\boldsymbol{D}_{33}\in\mathbb{R}^{q\times q}$要求是对角阵, $\boldsymbol{D}_{22}\in\mathbb{R}^{(n-p-q)\times(n-p-q)}$要求是对称不可约三对角阵, $\boldsymbol{D}_{11}\in\mathbb{R}^{p\times p}$. 如果 $q=n$, 则停止, 算法完成. 否则

③ 应用算法 8.5.2 到 $\boldsymbol{D}_{22}$.

$$\boldsymbol{D}=\mathrm{diag}(\boldsymbol{I}_p,\boldsymbol{J},\boldsymbol{I}_q)^{\mathrm{T}}\boldsymbol{D}\mathrm{diag}(\boldsymbol{I}_p,\boldsymbol{J},\boldsymbol{I}_q)$$

其中 $\boldsymbol{J}$ 为一系列吉文斯变换阵的乘积.

算法结束时, 得到对角阵 $\boldsymbol{D}$, 其对角线元素即为所求特征值. 无条件收敛且为三次收敛. 至于特征向量, 可用反幂法计算. 为节省存储, 该算法中的矩阵 $\boldsymbol{D}$ 和 $\boldsymbol{T}$ 均可覆盖矩阵 $\boldsymbol{A}$.

8.6 雅可比方法

雅可比方法是利用一系列吉文斯正交相似变换使实对称矩阵逐渐对角化的方法. 就收敛速度而言,雅可比法比 QR 法差得多. 近来引起人们注意,是因为它本质上是并行的. 此外,当矩阵本身已几乎是对角形或块对角形时,用雅可比方法特别有效,而且特征向量的计算也比较方便.

8.6.1 经典雅可比方法

设 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 是一个对称阵. 令 $S(\boldsymbol{A})=\sum\limits_{l\neq k}(a_{lk}^2)$ 表示 $\boldsymbol{A}$ 的非对角线元素的平方和, $D(\boldsymbol{A})=\sum\limits_{i=1}^{n}a_{ii}^2$ 表示 $\boldsymbol{A}$ 的对角线元素的平方和,则 $\boldsymbol{A}$ 的弗罗贝尼乌斯范数的平方 $\|\boldsymbol{A}\|_F^2=S(\boldsymbol{A})+D(\boldsymbol{A})$.

雅可比方法的思想是通过雅可比迭代逐步地减小 $S(\boldsymbol{A})$ 和增大 $D(\boldsymbol{A})$,实现此目的的工具是吉文斯变换. 记 $\boldsymbol{A}_0=\boldsymbol{A}$,雅可比迭代是完成一个正交相似变换过程,即

$$\boldsymbol{A}_k=\boldsymbol{J}_k^{\mathrm{T}}\boldsymbol{A}_{k-1}\boldsymbol{J}_k,\quad k=1,2,\cdots \tag{8.6.1}$$

其中 $\boldsymbol{J}_k$ 是吉文斯变换阵.

经典雅可比方法的基础是选择矩阵 $\boldsymbol{A}_{k-1}$ 中绝对值最大的非对角线元素 $a_{pq}^{(k-1)}(p\neq q,p<q)$,则有

$$\boldsymbol{J}_k=\boldsymbol{J}(p,q,\theta)=\begin{pmatrix}1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 1 & & & & & & \\ & & & \cos\theta & & & -\sin\theta & & \\ & & & & 1 & & & & \\ & & & & & \ddots & & & \\ & & & & & & 1 & & \\ & & & \sin\theta & & & \cos\theta & & \\ & & & & & & & 1 & \\ & & & & & & & & \ddots \\ & & & & & & & & & 1\end{pmatrix}\begin{matrix}\\ \\ \\ p\\ \\ \\ \\ q\\ \\ \\ \\ \end{matrix}$$

$$\qquad\qquad\qquad p \qquad\qquad\qquad q$$

显然，所有 $\boldsymbol{A}_k(k=1,2,\cdots)$ 都是对称阵，$\boldsymbol{A}_k$ 与 $\boldsymbol{A}_{k-1}$ 只是在第 p 和 q 这两行、两列的元素不相同，其余元素均没有变，且在正交相似变换下弗罗贝尼乌斯范数不变，即 $\|\boldsymbol{A}_k\|_F=\|\boldsymbol{A}_{k-1}\|_F$. 有如下关系：

$$\begin{cases} a_{ip}^{(k)}=a_{pi}^{(k)}=a_{ip}^{(k-1)}\cos\theta+a_{iq}^{(k-1)}\sin\theta \quad (i\neq p,q), \\ a_{iq}^{(k)}=a_{qi}^{(k)}=a_{iq}^{(k-1)}\cos\theta-a_{ip}^{(k-1)}\sin\theta \quad (i\neq p,q), \\ a_{pp}^{(k)}=a_{pp}^{(k-1)}\cos^2\theta+2a_{pq}^{(k-1)}\sin\theta\cos\theta+a_{qq}^{(k-1)}\sin^2\theta, \\ a_{qq}^{(k)}=a_{pp}^{(k-1)}\sin^2\theta+2a_{pq}^{(k-1)}\sin\theta\cos\theta+a_{qq}^{(k-1)}\cos^2\theta, \\ a_{pq}^{(k)}=a_{qp}^{(k)}=(a_{qq}^{(k-1)}-a_{pp}^{(k-1)})\sin\theta\cos\theta+a_{pq}^{(k-1)}(\cos^2\theta-\sin^2\theta) \end{cases} \tag{8.6.2}$$

选 θ，使

$$a_{pq}^{(k)}=a_{qp}^{(k)}=0. \tag{8.6.3}$$

易证 $(a_{ip}^{(k)})^2+(a_{iq}^{(k)})^2=(a_{ip}^{(k-1)})^2+(a_{iq}^{(k-1)})^2$，$i\neq p,q$，又因为 $\|\boldsymbol{A}_k\|_F=\|\boldsymbol{A}_{k-1}\|_F$，有

$$\begin{aligned}(a_{pp}^{(k-1)})^2+(a_{qq}^{(k-1)})^2+2a_{pq}^{(k-1)} &= (a_{pp}^{(k)})^2+(a_{qq}^{(k)})^2+2(a_{pq}^{(k)})^2 \\ &= (a_{pp}^{(k)})^2+(a_{qq}^{(k)})^2,\end{aligned}$$

从而有

$$D(\boldsymbol{A}_k)=D(\boldsymbol{A}_{k-1})+2(a_{pq}^{(k-1)})^2,$$
$$S(\boldsymbol{A}_k)=S(\boldsymbol{A}_{k-1})-2(a_{pq}^{(k-1)})^2.$$

因此，每作一次雅可比迭代，$D(\boldsymbol{A}_k)$ 增大了，$S(\boldsymbol{A}_k)$ 减小了，当 $k\to\infty$ 时，$S(\boldsymbol{A}_k)\to 0$，对于特征值的某个排列，$\boldsymbol{A}_k\to\mathrm{diag}(\lambda_i)$，且是二次收敛的.

由式(8.6.3)和式(8.6.2)的最后一个公式，推得

$$\tau=\tan 2\theta=2a_{pq}^{(k-1)}/(a_{pp}^{(k-1)}-a_{qq}^{(k-1)}). \tag{8.6.4}$$

满足上式的 θ 并不惟一确定，通常规定 θ 的范围

$$-\frac{\pi}{4}\leqslant\theta\leqslant\frac{\pi}{4}.$$

若 $a_{pp}^{(k-1)}-a_{qq}^{(k-1)}=0$，则取

$$\theta=\begin{cases}-\frac{\pi}{4}, & a_{pq}^{(k-1)}<0,\\ \frac{\pi}{4}, & a_{pq}^{(k-1)}>0.\end{cases}$$

设 $t=\tan\theta$，由式(8.6.4)得二次方程 $t^2+2\tau t-1=0$，它的两个根为 $t=-\tau\pm\sqrt{1+\tau^2}$. 选择其中较小的一个根，它可保证 $|\theta|\leqslant\frac{\pi}{4}$. 令 $s=\sin\theta, c=\cos\theta, t=\frac{s}{c}$，有

$$c=\frac{1}{\sqrt{1+t^2}},\quad s=tc.$$

雅可比方法对特征向量的计算是方便的，假定经 l 次迭代后，$\boldsymbol{A}_l$ 已近似对角化，即

$$\boldsymbol{A}_l=\boldsymbol{J}_l^{\mathrm{T}}\cdots\boldsymbol{J}_1^{\mathrm{T}}\boldsymbol{A}\boldsymbol{J}_1\cdots\boldsymbol{J}_l\approx\boldsymbol{D},$$

其中 $\boldsymbol{D}$ 为对角阵. 令 $\boldsymbol{V}_l=\boldsymbol{J}_1\cdots\boldsymbol{J}_l$，则有

$$\boldsymbol{A}\boldsymbol{V}_l\approx\boldsymbol{V}_l\boldsymbol{D}.$$

所以，$\boldsymbol{V}_l$ 的第 j 列就是对应于近似特征值 $a_{jj}^{(l)}$ 的近似特征向量($j=1,2,\cdots,n$).

算法 8.6.1 经典雅可比方法

给定对称阵 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 和容差 tol>0

(1) $\boldsymbol{V}=\boldsymbol{I}_n$(单位阵)；eps=tol $\|\boldsymbol{A}\|_F$

(2) 选择(p,q)，使 $|a_{pq}|=\max\limits_{i\neq j}|a_{ij}|$

① 计算 $\boldsymbol{J}(p,q,\theta)$ 的元素 c 和 s

如果 $\boldsymbol{A}(p,q)\neq0$，则

$\tau=(\boldsymbol{A}(q,q)-\boldsymbol{A}(p,p))/(2\boldsymbol{A}(p,q))$

如果 $\tau\geqslant0$，则 $t=\dfrac{1}{\tau+\sqrt{1+\tau^2}}$

否则 $t=-\dfrac{1}{-\tau+\sqrt{1+\tau^2}}$

$$c=\frac{1}{\sqrt{1+t^2}}$$

$s=tc$

否则

$c=1$

$s=0$

② $\mathbf{A}=\boldsymbol{J}(p,q,\theta)^{\mathrm{T}}\mathbf{A}\boldsymbol{J}(p,q,\theta)$

只需计算第 p 和 q 两行两列元素,即用式(8.6.2)前 4 个式子.

③ $\mathbf{V}=\mathbf{V}\boldsymbol{J}(p,q,\theta)$

只需计算 $\mathbf{V}$ 的第 p 和 q 两列的元素,即

$$\begin{cases} v_{kp} := v_{kp}c + v_{kq}s, \\ v_{kq} := -v_{kp}s + v_{kq}c, \end{cases} \quad (k=1,2,\cdots,n)$$

(3) 如果 $S(\mathbf{A})<\mathrm{eps}$,则终止

否则转到(2)

8.6.2 行循环雅可比方法

经典雅可比方法的每次迭代都要选取绝对值最大的元素,计算工作量大.改进的办法是将变换的顺序固定,一种合理的选择是逐行对每个非对角元素进行变换,即按下面各行的次序:

$$\begin{matrix} (1,2) & (1,3) & \cdots & (1,n) \\ & (2,3) & \cdots & (2,n) \\ & & \ddots & \vdots \\ & & & (n-1,n) \end{matrix}$$

这种排序格式称为按行循环,得到**行循环雅可比方法**.

算法 8.6.2 行循环雅可比方法

给定对称阵 $\mathbf{A}\in\mathbb{R}^{n\times n}$ 和容差 $\mathrm{tol}>0$

(1) $\mathbf{V}=\boldsymbol{I}_n$, $\mathrm{eps}=\mathrm{tol}\,\|\mathbf{A}\|_F$

(2) 对于$p=1,2,\cdots,n-1$

对于 $q=p+1,p+2,\cdots,n$

同算法 8.6.1 中第(2)步的①,②和③

(3) 如果 $S(\boldsymbol{A})<\text{eps}$,则终止

否则转到(2)

行循环雅可比方法也是二次收敛的.但收敛速度不如对称 QR 算法.

通过并行排序,可得到并行雅可比方法(略).

8.7 子空间迭代法

子空间迭代法(subspace iteration)可看做幂法的块推广,即同时用若干个线性无关的正交规范化向量进行幂迭代,又称它为**同时迭代法**(simultaneous iteration).迭代收敛后,同时得到矩阵的若干个特征值和相应的特征向量.它是结构工程中所用的最重要的方法,是目前求解大型稀疏矩阵特征值问题的最有效的方法之一.

设矩阵 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 的特征值$\lambda_i(i=1,2,\cdots,n)$排列为:

$$|\lambda_1|\geqslant|\lambda_2|\geqslant\cdots\geqslant|\lambda_n|,$$

假定要求 m 个占优的特征值和相应的特征向量.

算法 8.7.1 简单的子空间迭代

(1) 任取 m 个初始向量,将它们正交规范化,并以它们为列组成矩阵 $\boldsymbol{X}_0=(\boldsymbol{x}_1,\cdots,\boldsymbol{x}_m)\in\mathbb{R}^{n\times m}$

(2) 对于 $k=1,2,\cdots$迭代直到收敛

① 计算 $\boldsymbol{X}_k:=\boldsymbol{A}\boldsymbol{X}_{k-1}$

② 对 $\boldsymbol{X}_k\in\mathbb{R}^{n\times m}$作 QR 分解,即 $\boldsymbol{X}_k=\boldsymbol{Q}\boldsymbol{R}$

③ $\boldsymbol{X}_k:=\boldsymbol{Q}$

为减少步骤(2)中②的太频繁的正交规范化,改进的办法是在

步骤(2)中①执行若干次迭代后，再作正交规范化. 有如下的多步子空间迭代法.

算法 8.7.2 多步子空间迭代法

(1) 任选 m 个正交规范化向量，构成初始矩阵 $\boldsymbol{X}=(x_1, x_2, \cdots, x_m)\in\mathbb{R}^{n\times m}$ 和选定初始迭代参数 iter(正整数)

(2) 迭代直到收敛

① 计算 $\boldsymbol{z}=\boldsymbol{A}^{\text{iter}}\boldsymbol{X}$

② 正交规范化 $\boldsymbol{z}$，即 $\boldsymbol{z}=\boldsymbol{QR}$(QR 分解)

③ $\boldsymbol{X}=\boldsymbol{Q}$

④ 选新的 iter

如果在算法中采用投影过程，可得到更好的舒尔向量(见 8.2.1节)的近似. 有如下的带投影的子空间迭代法.

算法 8.7.3 带投影的子空间迭代法

(1) 同算法 8.7.2 中的(1)

(2) 迭代直到收敛

① 计算 $\hat{\boldsymbol{Z}}=\boldsymbol{A}^{\text{iter}}\boldsymbol{X}_{\text{old}}$

② 正交规范化 $\hat{\boldsymbol{Z}}$ 到 $\boldsymbol{Z}\in\mathbb{R}^{n\times m}$

③ 计算 $\boldsymbol{B}=\boldsymbol{Z}^{\mathrm{H}}\boldsymbol{AZ}$，其中 $\boldsymbol{B}\in\mathbb{R}^{m\times m}$

④ 用 QR 算法，计算 $\boldsymbol{B}$ 的舒尔向量组构成的矩阵 $\boldsymbol{Y}=(\boldsymbol{y}_1, \cdots, \boldsymbol{y}_m)$

⑤ 计算 $\boldsymbol{X}_{\text{new}}=\boldsymbol{ZY}$

⑥ 选新的迭代参数 iter.

迭代参数 iter 不能选得太大，否则步骤(2)中①的矩阵 $\hat{\boldsymbol{Z}}$ 的各列向量会几乎线性相关，这样步骤(2)中②的正交规范化会产生困难. 可从估计向量的收敛速度来确定 iter. 另外，$\|\boldsymbol{X}_{\text{new}}-\boldsymbol{X}_{\text{old}}\|<\epsilon$(预先给定的容差)可作为迭代终止的标准.

如果要了解更多的算法实施细节，如一旦第一个特征向量已收敛，如何**锁定**(locking)它，或**收缩**(deflation)，还有**位移**(shifts)

和预处理技术等,可参考有关资料.

如果实矩阵 $\boldsymbol{A}$ 的 m 个占优特征值满足

$$|\lambda_1|>|\lambda_2|>\cdots>|\lambda_m|,$$

令 $Q=(\boldsymbol{q}_1,\cdots,\boldsymbol{q}_m)$,它的各列是相应于 $\lambda_j(j=1,2,\cdots,m)$ 的舒尔向量,则 $\boldsymbol{X}$ 的第 j 个列向量收敛于 $\boldsymbol{q}_j(j=1,2,\cdots,m)$,其收敛速度取决于 $\left|\frac{\lambda_{j+1}}{\lambda_j}\right|$. 矩阵 $\boldsymbol{B}$ 收敛于上三角阵; 当 $\boldsymbol{A}$ 对称时,$\boldsymbol{B}$ 收敛于对角阵. $\boldsymbol{B}$ 的对角线元素为 $\boldsymbol{A}$ 的 m 个特征值 $\lambda_1,\lambda_2,\cdots,\lambda_m$.

如果矩阵 $\boldsymbol{A}$ 的 m 个占优特征值不是互不相同的,则 $\boldsymbol{B}$ 收敛于块三角阵; 当 $\boldsymbol{A}$ 对称时,$\boldsymbol{B}$ 收敛于块三对角阵. 其中 1×1 子块为 $\boldsymbol{A}$ 的实特征值,2×2 子块为 $\boldsymbol{A}$ 的一对共轭复特征值.

8.8 阿诺尔德方法

阿诺尔德过程(见 7.11.2 节)是建立克雷洛夫子空间(见 7.11.1节)标准正交基的算法. 最初用来约化一个稠密阵为海森伯格阵. 后来发现它也是求解大型稀疏矩阵近似特征值的好方法.

从任意非零规范化向量 $\boldsymbol{v}_1$ 开始,由幂迭代生成序列 $\boldsymbol{v}_k=\boldsymbol{A}\boldsymbol{v}_{k-1}$ $(k=1,2,\cdots,m)$,构成克雷洛夫子空间 K_m 的基,即 $K_m=\mathrm{span}\{\boldsymbol{v}_1, \boldsymbol{A}\boldsymbol{v}_1,\cdots,\boldsymbol{A}^{m-1}\boldsymbol{v}_1\}$. 用如下的阿诺尔德过程即用修正的格拉姆-施密特正交化方法将 K_m 的基正交化,称为阿诺尔德-**修正格拉姆-施密特算法**. 修正的格拉姆-施密特算法与标准的格拉姆-施密特算法在数学上等价,采用形式不同的计算公式,修正的算法比标准的算法具有更好的数值性质.

算法 8.8.1 阿诺尔德-修正格拉姆-施密特算法

(1) 给定一般的 $n\times n$ 非埃尔米特阵 $\boldsymbol{A}$,任选非零向量 $\boldsymbol{v}_1$ 且 $\|\boldsymbol{v}_1\|_2=1$. 确定 $m(<n)$

(2) 迭代

对于$j=1,2,\cdots,m$

① $\boldsymbol{\omega}:=\boldsymbol{A}\boldsymbol{v}_j$

② 对于 $i=1,2,\cdots,j$

$$h_{ij}=(\boldsymbol{\omega},\boldsymbol{v}_i)$$

$$\boldsymbol{\omega}:=\boldsymbol{\omega}-h_{ij}\boldsymbol{v}_i$$

③ $h_{j+1,j}=\|\boldsymbol{\omega}\|_2$

④ 如果 $h_{j+1,j}=0$,则停止

否则

⑤ $\boldsymbol{v}_{j+1}=\boldsymbol{\omega}/h_{j+1,j}$

该算法有如下性质:

命题 8.8.2 向量 $v_1,v_2,\cdots,\boldsymbol{v}_m$ 构成子空间 $\boldsymbol{K}_m=\{\boldsymbol{v}_1,\boldsymbol{A}\boldsymbol{v}_1,\cdots,\boldsymbol{A}^{m-1}\boldsymbol{v}_1\}$的正交基.

命题 8.8.3 令 $\boldsymbol{V}_m=(\boldsymbol{v}_1,\cdots,\boldsymbol{v}_m)$是 $n\times m$ 矩阵,$\boldsymbol{H}_m$ 是 $m\times m$ 海森伯格矩阵,其非零元素由算法确定则成立如下关系

$$\boldsymbol{A}\boldsymbol{V}_m=\boldsymbol{V}_m\boldsymbol{H}_m+h_{m+1,m}\boldsymbol{v}_{m+1}\boldsymbol{e}_m^{\mathrm{H}},\tag{8.8.1}$$

$$\boldsymbol{V}_m^{\mathrm{H}}\boldsymbol{A}\boldsymbol{V}_m=\boldsymbol{H}_m.\tag{8.8.2}$$

命题 8.8.4 令 $\lambda_i^{(m)}$ 是 $\boldsymbol{H}_m$ 的特征值(称为**里茨**(Ritz)值),相应的特征向量是 $\boldsymbol{y}_i^{(m)}$. $\boldsymbol{u}_i^{(m)}=\boldsymbol{V}_m\boldsymbol{y}_i^{(m)}$ 称为**里茨近似特征向量**,则

$$(\boldsymbol{A}-\lambda_i^{(m)}\boldsymbol{I})\boldsymbol{u}_i^{(m)}=h_{m+1,m}\boldsymbol{e}_m^{\mathrm{H}}\boldsymbol{y}_i^{(m)}\boldsymbol{v}_{m+1},\tag{8.8.3}$$

$$\|(\boldsymbol{A}-\lambda_i^{(m)}\boldsymbol{I})\boldsymbol{u}_i^{(m)}\|_2=h_{m+1,m}\mid\boldsymbol{e}_m^{\mathrm{H}}\boldsymbol{y}_i^{(m)}\mid.\tag{8.8.4}$$

该命题说明余量范数等于 $h_{m+1,m}$乘特征向量 $\boldsymbol{y}_i^{(m)}$ 的最后一个分量,从而得到终止准则. 当 $h_{j+1,j}=0$ 时,有

$$\boldsymbol{A}\boldsymbol{u}_i^{(j)}=\lambda_i^{(j)}\boldsymbol{u}_i^{(j)},i=1,\cdots,j,$$

这时 $\lambda_i^{(j)}$ 和 $\boldsymbol{u}_i^{(j)}$ 分别为原矩阵$\boldsymbol{A}$ 的最佳近似特征值和相应的特征向量.

为此,在完成了阿诺尔德-修正格拉姆-施密特算法之后,还需用其他方法,如海森伯格阵的 QR 算法(见 8.4.2 节)计算 $\boldsymbol{H}_m$ 的特征值$\lambda_i^{(m)}(i=1,2,\cdots,m)$和相应的特征向量 $\boldsymbol{y}_i^{(m)}(i=1,2,\cdots.$

m). 再计算 $\boldsymbol{u}_i^{(m)}=\boldsymbol{V}_m\boldsymbol{y}_i^{(m)}$ $(i=1,2,\cdots,m)$.

当 $m \ll n$ 时，很容易完成以上计算. 但随着 m 的增加，矩阵 $\boldsymbol{V}_m$ 和 $\boldsymbol{H}_m$ 的存储量增加，阿诺尔德算法的使用受限制，如果只需求出大型稀疏矩阵 $\boldsymbol{A}$ 的最大的实特征值，可利用重新开始的办法来应对 m 增大造成的困难. 在阿诺尔德迭代了 m 次后，计算 $\boldsymbol{A}$ 的近似特征向量，用它作为下一次阿诺尔德迭代的初始向量，如此重复，直到收敛. 有如下算法.

算法 8.8.5 迭代阿诺尔德算法

(1) 选初始向量 $\boldsymbol{v}_1 \in \mathbb{R}^n$ 且 $\|\boldsymbol{v}_1\|_2=1$. 选定矩阵 $\boldsymbol{V}_m$ 和 $\boldsymbol{H}_m$ 的维数 m.

(2) 迭代：执行算法 8.8.1 的迭代步骤(2).

(3) 重新开始：计算 $\boldsymbol{H}_m$ 的最大特征值 $\lambda_1^{(m)}$ 和相应的特征向量 $\boldsymbol{y}_1^{(m)}$，计算 $\boldsymbol{u}_1^{(m)}=\boldsymbol{V}_m\boldsymbol{y}_1^{(m)}$.

(4) 如果其余量范数足够小，则停止，否则 $\boldsymbol{v}_1=\boldsymbol{u}_1^{(m)}$ 且转到(2).

算法求得的 $\lambda_1^{(m)}$ 和 $\boldsymbol{u}_1^{(m)}$ 分别是原矩阵 $\boldsymbol{A}$ 的最大近似特征值和相应的近似特征向量.

该算法是求解大型稀疏矩阵极端(最大和最小)特征值的很有效的算法，其收敛速度比幂法或带位移的幂法快得多，而且可选用比较小的 $m<n$，仅需用很有限的存储量.

8.9 兰乔斯方法

设 $n\times n$ 矩阵 $\boldsymbol{A}$ 是大型稀疏埃尔米特阵(当 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 时，$\boldsymbol{A}$ 是对称阵)，则阿诺尔德方法就简化为**兰乔斯**(Lanczos)**方法**. 它是求解 $\boldsymbol{A}$ 的少数几个最大或最小特征值的有效方法.

设 $\boldsymbol{A}$ 是 n 阶实对称阵，则存在 n 阶的正交阵 $\boldsymbol{V}_n=(\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_n)$，其中 $\boldsymbol{v}_i\in\mathbb{R}^n$ $(i=1,2,\cdots,n)$，使得

$$V_n^{\mathrm{T}} A V_n = T_n = \begin{pmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{n-2} & \alpha_{n-1} & \beta_{n-1} \\ & & & \beta_{n-1} & \alpha_n \end{pmatrix}, \tag{8.9.1}$$

显然 $\boldsymbol{A V}_n = \boldsymbol{V}_n \boldsymbol{T}_n$，即

$$\boldsymbol{A}(\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_n) = (\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_n) \begin{pmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_{n-1} \\ & & \beta_{n-1} & \alpha_n \end{pmatrix}.$$

对于 $j=1,2,\cdots,n-1$，有

$$\boldsymbol{A v}_j = \beta_{j-1} \boldsymbol{v}_{j-1} + \alpha_j \boldsymbol{v}_j + \beta_j \boldsymbol{v}_{j+1}, \tag{8.9.2}$$

其中 $\beta_0 \boldsymbol{v}_0 = \boldsymbol{0}$.

兰乔斯过程是确定三对角阵 $\boldsymbol{T}_n$ 的元素 $\{\alpha_i, \beta_i\}$ 及正交阵 $\boldsymbol{V}_n$ 的各列 $\{\boldsymbol{v}_i\}$ 的算法.

取 $\boldsymbol{v}_1 \in \mathbb{R}^n$ 且 $\|\boldsymbol{v}_1\|_2 = 1$，由式(8.9.2)且利用 $\{\boldsymbol{v}_j\}$ 的正交性，得

$$\alpha_j = (\boldsymbol{A v}_j, \boldsymbol{v}_j).$$

$$\tag{8.9.3}$$

令 $$\boldsymbol{w}_j = \boldsymbol{A v}_j - \alpha_j \boldsymbol{v}_j - \beta_{j-1} \boldsymbol{v}_{j-1},$$

若 $\boldsymbol{w}_j \neq \boldsymbol{0}$，则

$$\boldsymbol{v}_{j+1} = \boldsymbol{w}_j / \beta_{j+1}, \quad \beta_{j+1} = \|\boldsymbol{w}_j\|_2.$$

若 $\boldsymbol{w}_j = \boldsymbol{0}$，则计算停止. 最后

$$\alpha_n = (\boldsymbol{A v}_n, \boldsymbol{v}_n).$$

算法 8.9.1 兰乔斯过程

给定对称阵 $\boldsymbol{A} \in \mathbb{R}^{n \times n}$

(1) 选初始向量 $\boldsymbol{v}_1 \in \mathbb{R}^n$ 且 $\|\boldsymbol{v}_1\|_2 = 1$，置 $\beta_1 = 0$，$\boldsymbol{v}_0 = \boldsymbol{0}$. 确定 $m(<n)$.

(2) 迭代：对于 $j=1,2,\cdots,m$

$$\boldsymbol{w}_j := \boldsymbol{A}\boldsymbol{v}_j - \beta_j \boldsymbol{v}_{j-1}$$
$$\alpha_j := (\boldsymbol{w}_j, \boldsymbol{v}_j)$$
$$\boldsymbol{w}_j := \boldsymbol{w}_j - \alpha_j \boldsymbol{v}_j$$

如果 $\boldsymbol{w}_j=\boldsymbol{0}$,则停止,否则

$$\beta_{j+1} := \|\boldsymbol{w}_j\|_2$$
$$\boldsymbol{v}_{j+1} := \boldsymbol{w}_j/\beta_{j+1}$$

该算法有如下性质:

定理 8.9.2 设 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$为对称阵,$\boldsymbol{v}_1\in\mathbb{R}^n$且$\|\boldsymbol{v}_1\|_2=1$,则兰乔斯过程 8.9.1 进行到第 $j=m$ 步终止,其中 m 取子空间 k_m 的秩即 $m=\mathrm{rank}K_m$, $K_m=\mathrm{span}\{\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_m\}$. 此外,对所有的 $j=1,2,\cdots,m$ 有下式成立:

$$\boldsymbol{A}\boldsymbol{V}_j = \boldsymbol{V}_j\boldsymbol{T}_j + \boldsymbol{w}_j\boldsymbol{e}_j^{\mathrm{T}}, \tag{8.9.4}$$

其中 $\boldsymbol{V}_j=(\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_m)$且

$$\boldsymbol{T}_j = \begin{pmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_{j-1} \\ & & \beta_{j-1} & \alpha_j \end{pmatrix}.$$

定理 8.9.3 令 $\lambda_i^{(m)}$ 是 T_m 的特征值,其相应的特征向量为 $\boldsymbol{y}_i^{(m)}$ 且 $\boldsymbol{u}_i^{(m)}=\boldsymbol{V}_m\boldsymbol{y}_i^{(m)}$ 为里茨近似特征向量,则有

$$\|(\boldsymbol{A}-\lambda_i^{(m)}\boldsymbol{I})\boldsymbol{u}_i^{(m)}\|_2 = \beta_m\,|\boldsymbol{e}_m^{\mathrm{T}}\boldsymbol{y}_i^{(m)}|.$$

此定理给出了 $\boldsymbol{T}_m$ 与 $\boldsymbol{A}$ 的特征值的一个可计算的误差界:

$$\min|\lambda_i-\lambda_i^{(m)}|\leqslant|\beta_m||\boldsymbol{e}_m^{\mathrm{T}}\boldsymbol{y}_i^{(m)}|,\quad i=1,2,\cdots,m,$$

其中 λ_i, $i=1,2,\cdots,m$ 为 $\boldsymbol{A}$ 的特征值.

关于兰乔斯过程的收敛性有如下结论:

设 $\boldsymbol{A}$ 为 n 阶对称阵,其特征值 $\lambda_1\geqslant\lambda_2\geqslant\cdots\geqslant\lambda_n$,相应的特征向量 $\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_n$, $\boldsymbol{T}_m$ 为兰乔斯过程第 m 步得到的三对角阵,其特征值为 $\lambda_1^{(m)}\geqslant\lambda_2^{(m)}\geqslant\cdots\geqslant\lambda_m^{(m)}$,则

$$\lambda_1 \geqslant \lambda_1^{(m)} \geqslant \lambda_1 - \frac{(\lambda_1 - \lambda_n)\tan^2(\phi_1)}{(C_{m-1}(1+2\rho_1))^2},$$

$$\lambda_n \leqslant \lambda_m^{(m)} \leqslant \lambda_n - \frac{(\lambda_1 - \lambda_n)\tan^2(\phi_n)}{(C_{m-1}(1+2\rho_n))^2},$$

其中 $\cos(\phi_1) = |(\boldsymbol{v}_1, \boldsymbol{u}_1)|$, $\cos(\phi_n) = |(\boldsymbol{v}_n, \boldsymbol{u}_n)|$, $\rho_1 = \frac{\lambda_1 - \lambda_2}{\lambda_2 - \lambda_n}$, $\rho_n = \frac{\lambda_{n-1} - \lambda_n}{\lambda_1 - \lambda_{n-1}}$, $C_{m-1}(x)$为 $m-1$ 次**切比雪夫**(Chebyshev)**多项式**.

当 m 增加,即 $C_{m-1}(x)$的次数增加,切比雪夫多项式在 $1+2\rho_1>1$ 及 $1+2\rho_n>1$ 处的值迅速增大,所以 $\boldsymbol{A}$ 的极端里茨值 $\lambda_1^{(m)}$ 和 $\lambda_m^{(m)}$ 能分别给出 $\boldsymbol{A}$ 的极端特征值 λ_1 和 λ_n 的很好的近似值,由此,得到如下的切比雪夫算法的基本步骤.

算法 8.9.4 切比雪夫算法

给定对称阵 $\boldsymbol{A} \in \mathbb{R}^{n \times n}$,计算 $\boldsymbol{A}$ 的最大和最小近似特征值.

(1) 选定 $m<n$.

(2) 执行兰乔斯过程(算法 8.9.1),得三对角阵 $\boldsymbol{T}_m$.

(3) 用对称 QR 算法 8.5.3 的步骤(2),计算 $\boldsymbol{T}_m$ 的特征值 $\lambda_1^{(m)} \geqslant \lambda_2^{(m)} \geqslant \cdots \geqslant \lambda_m^{(m)}$.

(4) 选定 s,使 $m<s<n$.

(5) 执行兰乔斯过程(算法 8.9.1),得三对角阵 $\boldsymbol{T}_s$.

(6) 用对称 QR 算法 8.5.3 的步骤(2),计算 $\boldsymbol{T}_s$ 的特征值 $\lambda_1^{(s)} \geqslant \lambda_2^{(s)} \geqslant \cdots \geqslant \lambda_s^{(s)}$.

(7) 如果 $|\lambda_1^{(m)} - \lambda_1^{(s)}|$ 和 $|\lambda_m^{(m)} - \lambda_s^{(s)}|$ 充分小,则停止,否则 $m := s$,返回(2).

切比雪夫算法是计算对称矩阵极端(最大和最小)特征值的有效方法.而且 $\lambda_2^{(m)} \approx \lambda_2$, $\lambda_3^{(m)} \approx \lambda_3$, $\lambda_{m-1}^{(m)} \approx \lambda_{n-1}$, $\lambda_{m-2}^{(m)} \approx \lambda_{n-2}$,但效果不如极端情况.

8.10 对称广义特征值问题

广义特征值问题(generalized eigenvalue problem)最常见的形式为

$$\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{B}\boldsymbol{x}. \tag{8.10.1}$$

如果存在非零向量 $\boldsymbol{x}$,使上式成立,则称 λ 为**广义特征值**,$\boldsymbol{x}$ 为相应的**特征向量**.

如果 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$是对称阵,$\boldsymbol{B} \in \mathbb{R}^{n\times n}$是对称正定阵,则称式(8.10.1)为**对称正定广义特征值问题**.

8.10.1 楚列斯基-对称 QR 算法

由于 $\boldsymbol{B}$ 是正定的,故存在楚列斯基因子分解

$$\boldsymbol{B} = \boldsymbol{L}\boldsymbol{L}^{\mathrm{T}},$$

其中 $\boldsymbol{L}$ 为实的非奇异的下三角阵. 于是式(8.10.1)等价于

$$\boldsymbol{L}^{-1}\boldsymbol{A}(\boldsymbol{L}^{-1})^{\mathrm{T}}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{x} = \lambda \boldsymbol{L}^{\mathrm{T}}\boldsymbol{x},$$

又等价于

$$\boldsymbol{C}\boldsymbol{y} = \lambda \boldsymbol{y} \tag{8.10.2}$$

其中 $\boldsymbol{C}=\boldsymbol{L}^{-1}\boldsymbol{A}(\boldsymbol{L}^{-1})^{\mathrm{T}}$,$\boldsymbol{y}=\boldsymbol{L}^{\mathrm{T}}\boldsymbol{x}$. 这是标准对称特征值问题. 下面给出一种算法,它既用了楚列斯基因子分解又用对称 QR 算法.

算法 8.10.1

(1) 计算矩阵 $\boldsymbol{B}$ 的楚列斯基因子分解 $\boldsymbol{B}=\boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$

(2) 计算 $\boldsymbol{C}=\boldsymbol{L}^{-1}\boldsymbol{A}(\boldsymbol{L}^{-1})^{\mathrm{T}}$

(3) 用对称 QR 算法,计算 $\boldsymbol{C}$ 的舒尔分解 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{C}\boldsymbol{Q}=\mathrm{diag}\,(a_1,\cdots,a_n)$

(4) 令 $\boldsymbol{X}=\boldsymbol{L}^{-\mathrm{T}}\boldsymbol{Q}$

算法中矩阵 $\boldsymbol{C}$ 可覆盖矩阵 $\boldsymbol{A}$. 矩阵 $\boldsymbol{C}$ 的近似特征值$\tilde{a}_1,\cdots,\tilde{a}_n$为广义特征值的近似值,$\boldsymbol{X}$ 的各列为相应的特征向量.

该算法的不足是:实际上许多对称正定矩阵特征值问题中,$\boldsymbol{A}$

和 $\boldsymbol{B}$ 都是大型稀疏矩阵，而 $\boldsymbol{C}$ 是满阵，它不能保持矩阵的稀疏性；另外当矩阵 $\boldsymbol{B}$ 是病态阵时，$\tilde{a}_i, i=1,\cdots,n$ 值严重损失有效位.

例 8.10.2 若 $\boldsymbol{A}=\begin{pmatrix}1 & 2 & 3\\ 2 & 4 & 5\\ 3 & 5 & 6\end{pmatrix}$，$\boldsymbol{L}=\begin{pmatrix}0.001 & 0 & 0\\ 1 & 0.001 & 0\\ 2 & 1 & 0.001\end{pmatrix}$ 及 $\boldsymbol{B}=\boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$，则 $\boldsymbol{A}-\lambda\boldsymbol{B}$ 的两个小特征值为

$$a_1=-0.619402940600584,$$

$$a_2=1.627440079051887.$$

若用算法 8.10.1，对该例作双精度运算，得

$$\tilde{a}_1=-0.619373517376444,$$

$$\tilde{a}_2=1.627516601905228.$$

只准确到 4 位有效数字，其原因在于矩阵 $\boldsymbol{B}$ 严重病态，其条件数 $\mathrm{cond}_2(\boldsymbol{B})\approx 10^{18}$.

8.10.2 兰乔斯算法

兰乔斯方法是求解高阶对称正定广义特征值问题的部分大的(或小的)特征值和相应的特征向量的有效方法.

对于标准对称特征值问题(8.10.2)，兰乔斯方法可保持矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 的稀疏性，且不必先将 $\boldsymbol{C}=\boldsymbol{L}^{-1}\boldsymbol{A}\boldsymbol{L}^{-\mathrm{T}}$ 乘出，而用其因子形式. 例如，算法中要计算 $\boldsymbol{z}=\boldsymbol{C}\boldsymbol{v}=\boldsymbol{L}^{-1}\boldsymbol{A}\boldsymbol{L}^{-\mathrm{T}}\boldsymbol{v}$，可作如下计算

$$\boldsymbol{u}=\boldsymbol{L}^{-\mathrm{T}}\boldsymbol{v},\quad \boldsymbol{w}=\boldsymbol{A}\boldsymbol{u},\quad \boldsymbol{z}=\boldsymbol{L}^{-1}\boldsymbol{w},$$

包含两个三角形方程组的求解和一个矩阵向量乘法. 因此，兰乔斯方法能求解高阶、带宽较大的对称正定广义特征值问题.

下面考虑与广义特征值问题(8.10.1)等价的另一种标准特征值问题

$$\boldsymbol{B}^{-1}\boldsymbol{A}\boldsymbol{x}=\lambda\boldsymbol{x}. \tag{8.10.3}$$

令 $\boldsymbol{C}=\boldsymbol{B}^{-1}\boldsymbol{A}$，因为 $\boldsymbol{A}$ 对称，$\boldsymbol{B}$ 对称正定，$\boldsymbol{B}^{-1}$ 显然是正定的，故 $\boldsymbol{C}$ 的

由于舍入误差,$(\boldsymbol{v},\boldsymbol{w})$有可能出现负值.一种改进的办法是:引进向量 $\boldsymbol{z}_j=\boldsymbol{B}\boldsymbol{v}_j$.在兰乔斯过程中对 $\boldsymbol{z}_j$ 保持三项递推关系,即用 $\boldsymbol{B}$ 左乘式(8.10.4)两边,得

$$\boldsymbol{B}\boldsymbol{w}=\boldsymbol{A}\boldsymbol{v}_j-\alpha_j\boldsymbol{z}_j-\beta_j\boldsymbol{z}_{j-1}.$$

这样,构成求解广义特征值问题的第二种兰乔斯过程.

算法 8.10.4 第二种兰乔斯过程

(1) 选初始向量$\boldsymbol{v}_1$ 且 $\|\boldsymbol{v}_1\|_B=1$,置 $\beta_1=0$,

$$\boldsymbol{z}_0=\boldsymbol{v}_0=\boldsymbol{0},\boldsymbol{z}_1=\boldsymbol{B}\boldsymbol{v}_1$$

(2) 迭代:对于 $j=1,2,\cdots,m$

① $\boldsymbol{v}:=\boldsymbol{A}\boldsymbol{v}_j-\beta_j\boldsymbol{z}_{j-1}$

② $\alpha_j:=(\boldsymbol{v},\boldsymbol{v}_j)$

③ $\boldsymbol{v}:=\boldsymbol{v}-\alpha_j\boldsymbol{z}_j$

④ $\boldsymbol{w}:=\boldsymbol{B}^{-1}\boldsymbol{v}$

⑤ $\beta_{j+1}:=(\boldsymbol{w},\boldsymbol{v})^{\frac{1}{2}}$

⑥ $\boldsymbol{v}_{j+1}=\boldsymbol{w}/\beta_{j+1}$ 和 $\boldsymbol{z}_{j+1}=\boldsymbol{v}/\beta_{j+1}$

因为 $\boldsymbol{B}$ 对称正定,故本算法第(2)步⑤中的$(\boldsymbol{w},\boldsymbol{v})=(\boldsymbol{B}^{-1}\boldsymbol{v},\boldsymbol{v})$的计算不会出现负值.

将第一种兰乔斯过程或第二种兰乔斯过程代替切比雪夫算法 8.9.4中的兰乔斯过程,可求得广义对称正定特征值问题的极端特征值.

特征值是实的且其规范化特征向量关于 $\boldsymbol{B}$ 内积

$$(\boldsymbol{x},\boldsymbol{y})_B=(\boldsymbol{Bx},\boldsymbol{y})$$

是正交的. 相应的 $\boldsymbol{B}$ 范数为

$$\|\boldsymbol{x}\|_B=(\boldsymbol{Bx},\boldsymbol{x})^{\frac{1}{2}}.$$

重要的是矩阵 $\boldsymbol{C}$ 关于欧几里得内积是非对称的,而关于 $\boldsymbol{B}$ 内积是自共轭的,即

$$(\boldsymbol{Cx},\boldsymbol{y})_B=(\boldsymbol{x},\boldsymbol{Cy})_B,\ \forall\,\boldsymbol{x},\boldsymbol{y}.$$

因此,可以对标准特征值问题(8.10.3)采用标准兰乔斯方法(见8.9节). 在兰乔斯过程(算法8.9.1)中,用 $\boldsymbol{B}$ 内积代替欧几里得内积,以及利用特征向量关于 $\boldsymbol{B}$ 内积正交的性质,对算法的主循环修改如下:

$$\begin{aligned}
&\alpha_j=(\boldsymbol{Cv}_j,\boldsymbol{v}_j)_B=(\boldsymbol{Av}_j,\boldsymbol{v}_j),\\
&\boldsymbol{w}:=\boldsymbol{Cv}_j-\alpha_j\boldsymbol{v}_j-\beta_j\boldsymbol{v}_{j-1},\\
&\beta_{j+1}=\|\boldsymbol{w}\|_B=(\boldsymbol{Bw},\boldsymbol{w})^{\frac{1}{2}}=(\boldsymbol{Av}_j,\boldsymbol{w})^{\frac{1}{2}}.
\end{aligned}\tag{8.10.4}$$

因为

$$(\boldsymbol{Bw},\boldsymbol{w})=(\boldsymbol{Av}_j,\boldsymbol{w})-\alpha_j(\boldsymbol{Bv}_j,\boldsymbol{w})-\beta_j(\boldsymbol{Bv}_{j-1},\boldsymbol{w})=(\boldsymbol{Av}_j,\boldsymbol{w}),$$

其中$(\boldsymbol{Bv}_j,\boldsymbol{w})=0$,$(\boldsymbol{Bv}_{j-1},\boldsymbol{w})=0$. 故有如下的求解广义特征值问题的第一种兰乔斯过程.

算法 8.10.3　第一种兰乔斯过程

(1) 选初始向量$\boldsymbol{v}_1$ 且 $\|\boldsymbol{v}_1\|_B=1$,置 $\beta_1=0$,$\boldsymbol{v}_0=\boldsymbol{0}$,确定 $m<n$.

(2) 迭代:对于 $j=1,2,\cdots,m$

① $\boldsymbol{v}:=\boldsymbol{Av}_j$

② $\alpha_j:=(\boldsymbol{v},\boldsymbol{v}_j)$

③ $\boldsymbol{w}:=\boldsymbol{B}^{-1}\boldsymbol{v}-\alpha_j\boldsymbol{v}_j-\beta_j\boldsymbol{v}_{j-1}$

④ $\beta_{j+1}:=(\boldsymbol{v},\boldsymbol{w})^{\frac{1}{2}}$

⑤ $\boldsymbol{v}_{j+1}:=\boldsymbol{w}/\beta_{j+1}$

9 非线性方程组数值解与最优化方法

9.1 引言

9.1.1 非线性方程组求解问题

用计算机求解非线性方程组的有效算法，对求解各种非线性力学问题、电路问题、经济平衡问题、非线性规划以及各种非线性偏微分方程离散化得到的方程组，都有重要的实际意义.

非线性方程组的一般表达式为

$$\begin{cases} f_1(x_1,\cdots,x_n)=0, \\ \quad\vdots \\ f_n(x_1,\cdots,x_n)=0, \end{cases} \tag{9.1.1}$$

其中 $n>1$，为正整数，$f_i(i=1,\cdots,n)$是定义在实 n 维空间$\mathbb{R}^n$中开域 D 上的实函数，若采用向量记号

$$\boldsymbol{F}(\boldsymbol{x})=\begin{pmatrix} f_1(\boldsymbol{x}) \\ \vdots \\ f_n(\boldsymbol{x}) \end{pmatrix},\quad \boldsymbol{x}=\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix},\quad \boldsymbol{0}=\begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix},$$

则方程(9.1.1)可写成

$$\boldsymbol{F}(\boldsymbol{x})=\boldsymbol{0}. \tag{9.1.2}$$

这里 $\boldsymbol{F}$ 表示定义在 D 上的非线性映射(mapping)，记为 $\boldsymbol{F}$: $D\subset\mathbb{R}^n\to\mathbb{R}^n$. 若存在 $\boldsymbol{x}^*\in D$ 使 $\boldsymbol{F}(\boldsymbol{x}^*)=\boldsymbol{0}$，则称 $\boldsymbol{x}^*$ 为方程(9.1.2)的解. 当 $n=1$ 时，就是方程求根问题(见第 5 章). 当方程(9.1.1)中所有 f_i 都是线性函数时，即为线性方程组；如果 f_i 中至少有一个是非线性函数，则称为非线性方程组. 当 $n>1$，$\boldsymbol{F}$ 为非线性时，方

程(9.1.1)的求解问题无论在理论上或实际求解方法上，均比上述两种情形困难得多，因为非线性方程组解的情况复杂，它可能无解，也可能有任意多个解.

例 9.1.1 $\begin{cases} f_1(x_1,x_2) = x_1^2 - x_2 + a = 0, \\ f_2(x_1,x_2) = -x_1 + x_2^2 + a = 0. \end{cases}$

这是 $n=2$ 的例题，其中实数 a 在 -1 与 1 之间变化，在 $\mathbb{R}^2$ 中每个方程代表平面上一条曲线，求方程组的解就是求两条曲线交点，若取 $a=1,\frac{1}{4},0,-1$，则四种情况的解见图 9.1.

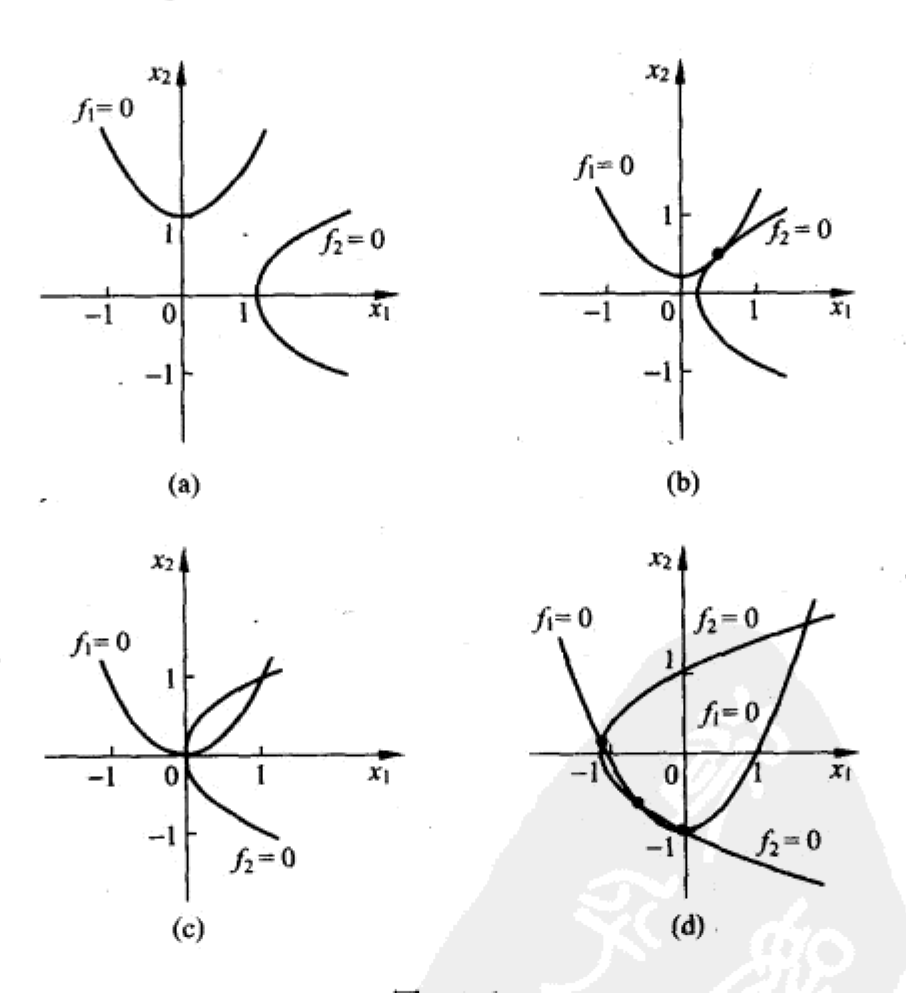


图 9.1

(a) $a=1$,无解;

(b) $a=\frac{1}{4}$,有惟一解,$x_1=x_2=\frac{1}{2}$;

(c) $a=0$,有两个解,$x_1=x_2=0$; $x_1=x_2=1$;

(d) $a=-1$,有 4 个解,$x_1=-1,x_2=0$; $x_1=0,x_2=-1$; $x_1=x_2=\frac{1}{2}(1\pm\sqrt{5})$.

例 9.1.2 方程组

$$\begin{cases} f_1(x_1,x_2)=\frac{1}{2}x_1\left[\sin\left(\frac{1}{2}\pi x_1\right)\right]-x_2=0, \\ f_2(x_1,x_2)=x_2^2-x_1+1=0, \end{cases}$$

有多个解,见图 9.2.

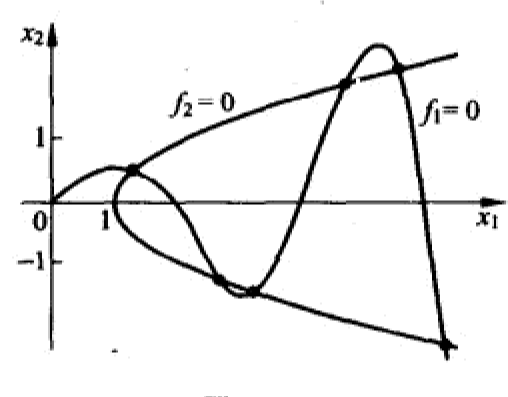


图 9.2

为求解非线性方程组(9.1.2),首先要知道解的情况,即有无解,有多少解;只要方程确实有解,就要研究描述、分析和逼近方程解的方法,主要是在计算机上使用较多的各种迭代算法和近十几年新发展的各种数值解法.

9.1.2 无约束最优化与非线性最小二乘

求解非线性方程组(9.1.1)可等价于无约束最优化问题,即求 $x^*\in D$,使

$$\varphi(\boldsymbol{x}^*) = \min_{\boldsymbol{x}\in D}\varphi(\boldsymbol{x}), \tag{9.1.3}$$

其中 φ: $D\subset\mathbb{R}^n\to\mathbb{R}^1$为给定的实函数，通常称为**目标函数**，$\boldsymbol{x}^*$ 称为问题(9.1.3)的最优解(或极小点). 问题(9.1.3)的含意是在函数的定义域 $D\subset\mathbb{R}^n$ 上求目标函数 $\varphi(\boldsymbol{x})$ 的极小值 $\boldsymbol{x}^*$，一个重要的特殊情形是在问题(9.1.3)中取

$$\varphi(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}^{\mathrm{T}}\boldsymbol{x} + c, \tag{9.1.4}$$

其中 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$为对称正定矩阵，$\boldsymbol{x},\boldsymbol{b}\in\mathbb{R}^n$，$c$ 为常数，此时称为**正定二次函数的极小问题**，它在最优化问题中起着重要作用.

如果目标函数 φ 可导，由多元函数必要条件，对 φ 求偏导数，则 $\boldsymbol{x}^*$ 是方程组

$$f_i(\boldsymbol{x}) = \frac{\partial\varphi(\boldsymbol{x})}{\partial \boldsymbol{x}_i} = 0 \quad (i = 1,2,\cdots,n) \tag{9.1.5}$$

的解，它表明求问题(9.1.3)的极小点可通过求解非线性方程组(9.1.5)得到. 反之，对方程组(9.1.2)，若取

$$\varphi(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{F}(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{F}(\boldsymbol{x}) = \sum_{i=1}^{n} f_i^2(x), \tag{9.1.6}$$

则可将求解非线性方程组(9.1.2)转化为求最优化问题(9.1.3).

在处理大量实验数据的曲线拟合问题时若数学模型关于参量不是线性的，假定拟合曲线方程为

$$y = f(t;\ \boldsymbol{x}), \tag{9.1.7}$$

其中 t 为自变量，$\boldsymbol{x}=(x_1,\cdots,x_n)^{\mathrm{T}}$ 为待定参量，若给出一组数据 $(t_i,y_i)(i=1,\cdots,m;\ m>n)$，要求 $\boldsymbol{x}$ 使

$$\varphi(\boldsymbol{x}) = \sum_{i=1}^{m}[f(t_i;\ \boldsymbol{x}) - y_i]^2 \tag{9.1.8}$$

最小，这里 $\boldsymbol{x}\in D\subset\mathbb{R}^n$，$\varphi\subset D\subset\mathbb{R}^n\to\mathbb{R}^1$，这就是**非线性最小二乘问题**. 它也是一个求最优解问题. 求式(9.1.8)的极小值，由多元函数极值必要条件得

$$\frac{\partial \varphi}{\partial x_j} = 2\sum_{i=1}^{m}[f(t_i; \boldsymbol{x}) - y_i]\frac{\partial}{\partial x_j}f(t_i;\boldsymbol{x}) = 0 \quad (j = 1,2,\cdots,n). \tag{9.1.9}$$

式(9.1.9)是关于 $\boldsymbol{x}$ 的非线性方程组,求此方程组的解与求式(9.1.8)的极小值是等价的.

此外,对非线性超定方程组

$$\begin{cases} f_1(x_1,\cdots,x_n) = 0, \\ \quad\vdots \\ f_m(x_1,\cdots,x_n) = 0, \end{cases} \quad (m > n) \tag{9.1.10}$$

的求解问题,当 $m=n$ 时,即为非线性方程组(9.1.1)求解,对于 $m>n$ 的超定方程组(9.1.10)的求解也可转化为求目标函数

$$\varphi(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{F}(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{F}(\boldsymbol{x}) = \frac{1}{2}\sum_{i=1}^{m}f_i^2(x_1,\cdots,x_n) \quad (m > n) \tag{9.1.11}$$

的极小值.因此,超定方程组(9.1.10)求解可通过求式(9.1.11)的最小二乘解得到.

9.1.3 非线性映射的导数与中值定理

非线性方程组(9.1.2)的求解涉及到映射 $\boldsymbol{F}: D\subset\mathbb{R}^n\to\mathbb{R}^m$ 的导数概念及相关性质.

定义 9.1.3 假定 $\boldsymbol{F}: D\subset\mathbb{R}^n\to\mathbb{R}^m$,如果对任何 $\boldsymbol{h}\in\mathbb{R}^n$, $\boldsymbol{x}+\boldsymbol{h}\in D$,都存在矩阵 $\boldsymbol{A}\in\mathbb{R}^{n\times m}$,使

$$\lim_{t\to 0}\frac{1}{t}\left\|\boldsymbol{F}(\boldsymbol{x}+t\boldsymbol{h}) - \boldsymbol{F}(\boldsymbol{x}) - t\boldsymbol{A}\boldsymbol{h}\right\| = 0, \tag{9.1.12}$$

或等价地

$$\lim_{t\to 0}\frac{1}{t}[\boldsymbol{F}(\boldsymbol{x}+t\boldsymbol{h}) - \boldsymbol{F}(\boldsymbol{x})] = \boldsymbol{A}\boldsymbol{h}, \tag{9.1.13}$$

则称映射 $\boldsymbol{F}$ 在 $\boldsymbol{x}$ 处 G-可导(Gateaux 可导),并称 $\boldsymbol{A}$ 为 $\boldsymbol{F}$ 在 $\boldsymbol{x}$ 处的

G-导数. 记为 $F'(\boldsymbol{x})$.

根据定义，当 $\boldsymbol{F}$ 在 $\boldsymbol{x}$ 处 G-可导时，则在 $\boldsymbol{x}$ 处 $\boldsymbol{F}$ 的各分量 $f_1,\cdots,f_m$ 的偏导数 $\frac{\partial f_i}{\partial x_j}(i=1,\cdots,m;\ j=1,\cdots,n)$ 均存在，记 $\boldsymbol{F}'(\boldsymbol{x})=\boldsymbol{A}=(a_{ij})$，取 $\boldsymbol{h}=\boldsymbol{e}_j=(0,\cdots,0,1,0,\cdots,0)^{\mathrm{T}}$ 为第 j 个单位向量，从式(9.1.13)得出

$$\lim_{t\to 0}\frac{1}{t}[f_i(\boldsymbol{x}+t\boldsymbol{e}^j)-f_i(\boldsymbol{x})]=a_{ij}.$$

由此得

$$a_{ij}=\frac{\partial f_i(\boldsymbol{x})}{\partial x_j},$$

即

$$\boldsymbol{F}'(\boldsymbol{x})=\begin{pmatrix}\frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n}\\ \vdots & & \vdots\\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n}\end{pmatrix}. \qquad (9.1.14)$$

矩阵(9.1.14)称为 $\boldsymbol{F}(\boldsymbol{x})$的雅可比矩阵.

当 $m=1$ 时 $F(\boldsymbol{x})$为定义于$\mathbb{R}^n$上的多元函数，$F(\boldsymbol{x})\equiv f(\boldsymbol{x})$，此时 $f'(\boldsymbol{x})=\left(\frac{\partial f(\boldsymbol{x})}{\partial x_1},\cdots,\frac{\partial f(\boldsymbol{x})}{\partial x_n}\right)$，相应地，$f'(\boldsymbol{x})^{\mathrm{T}}$ 称为 $f(\boldsymbol{x})$在点 $\boldsymbol{x}$ 的梯度，记 $\mathbf{grad}\,f(\boldsymbol{x})=f'(\boldsymbol{x})^{\mathrm{T}}$.

注意，G-导数是按方向定义的，$\boldsymbol{F}(\boldsymbol{x})$在 $\boldsymbol{x}$ 处 G-可导不能保证 $\boldsymbol{F}$ 在 $\boldsymbol{x}$ 处连续. 下面给出强导数的定义.

定义 9.1.4 假定 $\boldsymbol{F}$: $D\subset\mathbb{R}^n\to\mathbb{R}^m$，如果对 $\forall\,\boldsymbol{h}\in\mathbb{R}^n$，存在 $\boldsymbol{A}\in\mathbb{R}^{n\times m}$，使

$$\lim_{\boldsymbol{h}\to 0}\frac{1}{\|\boldsymbol{h}\|}\|\boldsymbol{F}(\boldsymbol{x}+\boldsymbol{h})-\boldsymbol{F}(\boldsymbol{x})-\boldsymbol{A}\boldsymbol{h}\|=0, \quad (9.1.15)$$

则称 $\boldsymbol{F}$ 在 $\boldsymbol{x}$ 处 F-**可导**(Frechet 可导). $\boldsymbol{A}$ 称为 F-**导数或强导数**.

仍记为 $\boldsymbol{A}=\boldsymbol{F}'(\boldsymbol{x})$.

显然,$\boldsymbol{F}$ 在 $\boldsymbol{x}$ 处 F-可导则必然 G-可导,且 F-导数等于 G-导数. $\boldsymbol{F}$ 在 $\boldsymbol{x}$ 处 F-可导则 $\boldsymbol{F}$ 连续.

有关导数最常用的定理是各种形式的中值定理.

定理 9.1.5 若 $\boldsymbol{F}$: $D\subset\mathbb{R}^n\to\mathbb{R}^m$ 在开凸集 $D_0\subset D$ 上 G-可导,对 $\forall\, \boldsymbol{x},\boldsymbol{y},\boldsymbol{z}\in D$,则有以下结论:

(1) 存在 $0\leqslant t_1,\cdots,t_m\leqslant 1$ 使

$$\boldsymbol{F}(\boldsymbol{y})-\boldsymbol{F}(\boldsymbol{x})=\begin{pmatrix} f'_1(\boldsymbol{x}+t_1(\boldsymbol{y}-\boldsymbol{x})) \\ \vdots \\ f'_m(\boldsymbol{x}+t_m(\boldsymbol{y}-\boldsymbol{x})) \end{pmatrix}(\boldsymbol{y}-\boldsymbol{x}). \tag{9.1.16}$$

(2)
$$\|\boldsymbol{F}(\boldsymbol{y})-\boldsymbol{F}(\boldsymbol{x})\| \leqslant \sup_{0\leqslant t\leqslant 1}\|\boldsymbol{F}'(\boldsymbol{x}+t(\boldsymbol{y}-\boldsymbol{x}))\|\,\|\boldsymbol{y}-\boldsymbol{x}\|. \tag{9.1.17}$$

(3)
$$\|\boldsymbol{F}(\boldsymbol{y})-\boldsymbol{F}(\boldsymbol{z})-\boldsymbol{F}'(\boldsymbol{x})(\boldsymbol{y}-\boldsymbol{z})\| \leqslant \sup_{0\leqslant t\leqslant 1}\|\boldsymbol{F}'(\boldsymbol{z}+t(\boldsymbol{y}-\boldsymbol{z}))-\boldsymbol{F}'(\boldsymbol{x})\|\,\|\boldsymbol{y}-\boldsymbol{z}\|. \tag{9.1.18}$$

(4) 若再假定 $\boldsymbol{F}'(\boldsymbol{x})$ 在 D_0 上半连续,则

$$\boldsymbol{F}(\boldsymbol{y})-\boldsymbol{F}(\boldsymbol{x})=\int_0^1\boldsymbol{F}'(\boldsymbol{x}+t(\boldsymbol{y}-\boldsymbol{x}))(\boldsymbol{y}-\boldsymbol{x})\mathrm{d}\boldsymbol{x}. \tag{9.1.19}$$

定理 9.1.6 若 $\boldsymbol{F}$: $D\subset\mathbb{R}^n\to\mathbb{R}^m$ 在开凸集 $D_0\subset D$ 上 G-可导,且对所有 $\boldsymbol{x},\boldsymbol{y}\in D_0$,若存在 $c>0$ 及 $p\in(0,1)$,使得

$$\|\boldsymbol{F}'(\boldsymbol{x})-\boldsymbol{F}'(\boldsymbol{y})\|\leqslant c\|\boldsymbol{x}-\boldsymbol{y}\|^p \tag{9.1.20}$$

成立,此时称 $\boldsymbol{F}'(\boldsymbol{x})$ 在 D_0 上为赫尔德连续,则

$$\|\boldsymbol{F}(\boldsymbol{y})-\boldsymbol{F}(\boldsymbol{x})-\boldsymbol{F}'(\boldsymbol{x})(\boldsymbol{y}-\boldsymbol{x})\|\leqslant\frac{c}{p+1}\|\boldsymbol{y}-\boldsymbol{x}\|^{p+1}. \tag{9.1.21}$$

9.2 迭代法与不动点定理

9.2.1 迭代法基本概念

为了研究迭代法,可把方程(9.1.2)改写成

$$\boldsymbol{x} = \boldsymbol{G}(\boldsymbol{x}), \tag{9.2.1}$$

这里 $\boldsymbol{G}: D\subset\mathbb{R}^n\to\mathbb{R}^n$. 例如 $\boldsymbol{G}(\boldsymbol{x})=\boldsymbol{x}+\boldsymbol{BF}(\boldsymbol{x})$,其中 $\boldsymbol{B}$ 为 n 阶非奇异矩阵,记 $\boldsymbol{B}\in\mathbb{R}^{n\times n}$. 方程(9.1.2)的解 $\boldsymbol{x}^*$,也是方程(9.2.1)的解,即 $\boldsymbol{x}^*=\boldsymbol{G}(\boldsymbol{x}^*)$,$\boldsymbol{x}^*$ 称为 G 的**不动点**(fixed point). 因此,求方程(9.2.1)的不动点等价于求方程(9.1.2)的解. 求方程解的迭代法的过程,就是从 $p\geqslant 1$ 个初始向量 $\boldsymbol{x}^0,\boldsymbol{x}^1,\cdots,\boldsymbol{x}^{p-1}$ 出发,通过某一迭代程序生成一个序列 $\{\boldsymbol{x}^k\}$ 的过程. 例如,可构造迭代程序

$$\boldsymbol{x}^{k+1} = \boldsymbol{G}(\boldsymbol{x}^k) \quad (k = 0,1,\cdots), \tag{9.2.2}$$

它由 $\boldsymbol{x}^0$ 出发生成一个序列 $\{\boldsymbol{x}^k\}$,此时 $p=1$,故称迭代法(9.2.2)为**单步法**(one-step methods). 当 $p>1$ 时,迭代程序为

$$\boldsymbol{x}^{k+1} = \boldsymbol{G}(x^k,x^{k-1}\cdots,x^{k-p+1}) \quad (k = p-1,p,\cdots), \tag{9.2.3}$$

称为**多步法**(multistep methods). 应用迭代法有三个基本问题:

第一,是适定性问题. 迭代法生成的序列 $\{\boldsymbol{x}^k\}$ 均在求解域 D 内,即 $\{\boldsymbol{x}^k\}\subset D$. 例如,对迭代法(9.2.2)就要求每步 $\boldsymbol{x}^k$ 均能计算出结果,且所有 $\boldsymbol{x}^k\in D$.

第二,是收敛性问题. 就是要求序列 $\{\boldsymbol{x}^k\}$ 收敛到方程(9.1.2)的解,即 $\lim\limits_{k\to\infty}\boldsymbol{x}^k=\boldsymbol{x}^*$. 由于对初始近似 $\boldsymbol{x}^0$ 的不同限制,迭代法收敛性有三种不同的概念.

(1) 局部收敛性.

定义 9.2.1 假定 $\boldsymbol{F}: D\subset\mathbb{R}^n\to\mathbb{R}^n$,$\boldsymbol{x}^*\in D$ 是方程(9.1.2)的一个解. 若存在 $\boldsymbol{x}^*$ 的一个邻域 $S\subset D$,使 $\forall\,\boldsymbol{x}^0\in S$,迭代序列 $\{\boldsymbol{x}^k\}\subset S$,且收敛于 $\boldsymbol{x}^*$,则称迭代序列 $\{\boldsymbol{x}^k\}$ 具有**局部收敛性**(local

convergence)，点 x^* 称为迭代序列 $\{x^k\}$ 的**吸引点**(point of attraction).

(2) 半局部收敛性. 它不假定方程(9.1.2)解 x^* 存在，只在初始近似 x^0 满足一定条件下证明迭代序列 $\{x^k\}$ 收敛于解 x^*，这种收敛性定理本身包含了证明解的存在惟一性. 实际上这时 x^0 仍在 x^* 附近，即 $\|x^0-x^*\|$ 较小，因此，半局部收敛性定理对 x^0 的限制仍然较大.

(3) 大范围收敛性. 它对求解域 D 中任意的初始近似 x^0，均能使序列 $\{x^k\}$ 收敛到解 x^*，是最完善的，也是最好的结果，但很多迭代法往往只具有局部或半局部收敛性.

第三，是效率问题. 在用计算机求方程解的全过程中，机时是否节省，这是一个时间计算复杂性问题，对非线性方程组迭代法可用迭代序列 $\{x^k\}$ 的收敛速度与每步迭代计算量大小来衡量.

定义 9.2.2 假定迭代序列 $\{x^k\}$ 收敛于 x^*，若存在实数 $p\geqslant 1$，使当 $k\geqslant k_0$，$x^k\neq x^*$ 时

$$0<\alpha_p=\lim_{k\to\infty}\frac{\|x^{k+1}-x^*\|}{\|x^k-x^*\|^p}<+\infty$$

成立，就说序列 $\{x^k\}$ 是 p **阶收敛**(order p convergence)，α_p 称为**收敛因子**(convergence factors). 特别地，$p=1$ 时，称为**线性收敛**(linear convergence)，$p>1$ 时，称为**超线性收敛**(superlinear convergence)，$p=2$ 时，称为**平方收敛**(quadratic convergence).

根据定义，一个迭代序列 $\{x^k\}$ 的收敛阶越高说明它收敛越快，但效率高低还与每步迭代的计算量有关.

定义 9.2.3 假定迭代序列 $\{x^k\}$ 的收敛阶为 $p\geqslant 1$，每步迭代的计算量为 w，则迭代序列的**效率**(efficiency) e 定义为

$$e=\frac{\ln p}{w}. \tag{9.2.4}$$

对线性收敛($p=1$)的迭代序列，效率 e 可定义为 $e=-\frac{\ln\alpha_1}{w}$，这里

$0<\alpha_1<1$ 为收敛因子.

9.2.2 压缩映射原理与不动点定理

对于方程(9.2.1)的解,其存在惟一性及迭代序列收敛性有以下定义与定理.

定义 9.2.4 假定 $G: D\subset\mathbb{R}^n\to\mathbb{R}^n$,若存在常数 $0<\alpha<1$,使 $\forall\, \boldsymbol{x},\boldsymbol{y}\in D_0\subset D$,有

$$\|G(\boldsymbol{x})-G(\boldsymbol{y})\|\leqslant\alpha\|\boldsymbol{x}-\boldsymbol{y}\| \qquad (9.2.5)$$

成立,则称 G 在 D_0 上为**压缩映射**(contraction mapping),α 称为**压缩系数**.

若对 $\forall\, \boldsymbol{x},\boldsymbol{y}\in D_0$ 有

$$\|G(\boldsymbol{y})-G(\boldsymbol{x})\|\leqslant\|\boldsymbol{y}-\boldsymbol{x}\|, \qquad (9.2.6)$$

则称 G 在 D_0 上为**非膨胀**(nonexpansive)**映射**,如果式(9.2.6)中当 $\boldsymbol{x}\neq\boldsymbol{y}$ 时严格不等式成立,则称 G 在 D_0 上为**严格非膨胀映射**.

定理 9.2.5(压缩映射原理) 设 $G: D\subset\mathbb{R}^n\to\mathbb{R}^n$ 为闭集 $D_0\subset D$ 上的压缩映射,且 $GD_0\subset D_0$,则 G 在 D_0 中存在惟一的不动点 $\boldsymbol{x}^*$. 而且,从任何 $\boldsymbol{x}^0\in D_0$ 出发,由迭代法(9.2.2)生成的序列 $\{\boldsymbol{x}^k\}$ 均收敛于 $\boldsymbol{x}^*$. 并有误差估计

$$\|\boldsymbol{x}^k-\boldsymbol{x}^*\|\leqslant\frac{\alpha^k}{1-\alpha}\|\boldsymbol{x}^1-\boldsymbol{x}^0\|.$$

定理 9.2.5 给出了方程(9.2.1)解存在惟一性的条件,只有在这个条件下,迭代法(9.2.2)是大范围收敛的. 但在实际计算时,构造的迭代法往往不能满足定理 9.2.5 的条件,而只在解 $\boldsymbol{x}^*$ 附近满足定理条件,即条件较弱的局部收敛定理.

定理 9.2.6 设 $\boldsymbol{x}^*$ 是方程(9.2.1)的解,$G: D\subset\mathbb{R}^n\to\mathbb{R}^n$,若存在开球 $S=\{\boldsymbol{x}\mid\|\boldsymbol{x}-\boldsymbol{x}^*\|<\delta,\delta>0\}=S(\boldsymbol{x}^*,\delta)\subset D$ 和常数 $0<\alpha<1$,使 $\forall\,\boldsymbol{x}\in S$,有

$$\|G(\boldsymbol{x})-G(\boldsymbol{x}^*)\|\leqslant\alpha\|\boldsymbol{x}-\boldsymbol{x}^*\|, \qquad (9.2.7)$$

可得

$$\mathbf{0} = \boldsymbol{F}(\boldsymbol{x}^*) = \boldsymbol{F}(\boldsymbol{x}^k) + \boldsymbol{F}'(\boldsymbol{x}^k)(\boldsymbol{x}^* - \boldsymbol{x}^k) + R(\boldsymbol{x}^* - \boldsymbol{x}^k),$$

其中

$$\lim_{\|\Delta x\| \to 0} \frac{\|\boldsymbol{R}(\Delta x)\|}{\|\Delta \boldsymbol{x}\|} = 0.$$

因此，当 $\boldsymbol{x}^k$ 靠近 $\boldsymbol{x}^*$ 时，可略去余项 $R(\boldsymbol{x}^* - \boldsymbol{x}^k)$，从而用线性方程组

$$\boldsymbol{F}'(\boldsymbol{x}^k)\Delta \boldsymbol{x} = -\boldsymbol{F}(\boldsymbol{x}^k) \tag{9.3.1}$$

的解 $\Delta \boldsymbol{x}$ 近似差 $\boldsymbol{x}^* - \boldsymbol{x}^k$，记 $\Delta \boldsymbol{x} = \boldsymbol{x}^{k+1} - \boldsymbol{x}^k$. 于是可得

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \Delta \boldsymbol{x} = \boldsymbol{x}^k - \boldsymbol{F}'(\boldsymbol{x}^k)^{-1}\boldsymbol{F}(\boldsymbol{x}^k).$$

作为 $\boldsymbol{x}^*$ 的新近似，可得到迭代程序：

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \boldsymbol{F}'(\boldsymbol{x}^k)^{-1}\boldsymbol{F}(\boldsymbol{x}^k) \quad (k = 0,1,\cdots), \tag{9.3.2}$$

此即为解非线性方程组的**牛顿法**，这里 $\boldsymbol{F}'(\boldsymbol{x})$ 是 $\boldsymbol{F}$ 的雅可比矩阵(9.1.14). 公式(9.3.2)只是一种形式记号，实际计算时求逆就是解方程(9.3.1)，故可将式(9.3.2)改为下列形式：

$$\begin{cases} \boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \Delta \boldsymbol{x}^k, \\ \boldsymbol{F}'(\boldsymbol{x}^k)\Delta \boldsymbol{x}^k = -\boldsymbol{F}(\boldsymbol{x}^k) \quad (k = 0,1,\cdots). \end{cases} \tag{9.3.3}$$

牛顿法由 $\boldsymbol{x}^k$ 计算 $\boldsymbol{x}^{k+1}$ 的步骤是：

(1) 计算 $\boldsymbol{F}(\boldsymbol{x}^k)$ 及 $\boldsymbol{F}'(\boldsymbol{x}^k)$；

(2) 解线性方程组 $\boldsymbol{F}'(\boldsymbol{x}^k)\Delta \boldsymbol{x}^k = -\boldsymbol{F}(\boldsymbol{x}^k)$，求得 $\Delta \boldsymbol{x}^k$；

(3) 令 $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \Delta \boldsymbol{x}^k$.

例 9.3.1 考虑方程组

$$\begin{cases} f_1(x_1,x_2) = 4x_1^2 + x_2^2 - 4 = 0, \\ f_2(x_1,x_2) = x_1 + x_2 - \sin(x_1 - x_2) = 0, \end{cases}$$

在 $(x_1^0, x_2^0) = (1,0)$ 附近的解.

解 求 $\boldsymbol{F}$ 的雅可比矩阵

$$\boldsymbol{F}'(\boldsymbol{x}) = \begin{bmatrix} \dfrac{\partial f_1}{\partial x_1} & \dfrac{\partial f_1}{\partial x_2} \\ \dfrac{\partial f_2}{\partial x_1} & \dfrac{\partial f_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 8x_1 & 2x_2 \\ 1 - \cos(x_1 - x_2) & 1 + \cos(x_1 - x_2) \end{bmatrix},$$

则 $\forall x^0 \in S$，由迭代法(9.2.2)生成的序列 $\{x^k\}$ 仍在 S 中，且收敛于 x^*.

定理 9.2.6 给出的条件(9.2.7)，实际上是 x^* 点的压缩条件，它不易检验，如果 G 在 x^* 处可导，则可得到下面的定理.

定理 9.2.7 设映射 $G: D \subset \mathbb{R}^n \to \mathbb{R}^n$ 有不动点 $x^* \in \mathrm{int}(D)$，且 G 在 x^* 处 F-可导，$G'(x^*)$ 的谱半径

$$\rho(G'(x^*)) = \sigma < 1, \tag{9.2.8}$$

则存在开球 $S = S(x^*, \delta) = \{x \in \mathbb{R}^n \mid \| x - x^* \| \leqslant \delta\} \subset D$，对任意初始近似 $x^0 \in S$，x^* 是迭代序列(9.2.2)的吸引点.

这个定理给出的条件可作为判断具体迭代序列收敛的依据.

局部收敛性是在 G 的不动点 x^* 存在的前提下得到的，当 G 不满足压缩条件(9.2.5)，而只是严格非膨胀映射时，有下列不动点定理.

定理 9.2.8 设 $G: D \subset \mathbb{R}^n \to \mathbb{R}^n$ 为有界闭集 $D_0 \subset D$ 上的严格非膨胀映射，且 $G(D_0) \subset D$. 则 G 在 D_0 上存在惟一不动点.

定理 9.2.9 设 $G: D \subset \mathbb{R}^n \to \mathbb{R}^n$ 在凸闭集 $D_0 \subset D$ 上是非膨胀映射，且 $G(D_0) \subset D$. 则 G 在 D_0 中存在不动点的充分必要条件是至少有一个 $x^0 \in D_0$，使序列 $x^{k+1} = G(x^k)\ (k = 0, 1, \cdots)$ 有界.

关于不动点存在性，最著名的是下列定理.

定理 9.2.10 (Brouwer 不动点定理) 设 $G: D \subset \mathbb{R}^n \to \mathbb{R}^n$ 在有界凸闭集 $D_0 \subset D$ 上连续，且 $G(D_0) \subset D_0$，则 G 在 D_0 上至少存在一个不动点.

9.3 牛顿型方法

9.3.1 牛顿法与牛顿型方法

求解方程(9.1.1)的牛顿法是 $n=1$ 的牛顿法(5.4.3)的推广，假定 x^* 是方程的解，它的 k 次近似 $x^k \in D$，利用多元泰勒展开，

$\mathbb{R}^n$在 $\boldsymbol{x}^*$ 处 F-可导，n 阶矩阵 $\boldsymbol{A}(\boldsymbol{x})$ 在 $\boldsymbol{x}^*$ 的邻域 S_0 内有定义，且在 $\boldsymbol{x}^*$ 处连续，$[\boldsymbol{A}(\boldsymbol{x}^*)]^{-1}$ 存在，则存在闭球 $S=\bar{S}(x^*,\delta)\subset S_0$，使对任何 $x\in S$，映射

$$\boldsymbol{G}(\boldsymbol{x}) = \boldsymbol{x} - [\boldsymbol{A}(\boldsymbol{x})]^{-1}\boldsymbol{F}(\boldsymbol{x}) \tag{9.3.8}$$

有定义，且在 $\boldsymbol{x}^*$ 处有 F-导数：

$$\boldsymbol{G}'(\boldsymbol{x}^*) = \boldsymbol{I} - [\boldsymbol{A}(\boldsymbol{x}^*)]^{-1}\boldsymbol{F}'(\boldsymbol{x}^*). \tag{9.3.9}$$

若还有

$$\rho(\boldsymbol{I} - [\boldsymbol{A}(\boldsymbol{x}^*)]^{-1}\boldsymbol{F}'(\boldsymbol{x}^*)) < 1,$$

则 $\boldsymbol{x}^*$ 是牛顿型迭代序列(9.3.5)的吸引点.

根据此定理，若取 $\boldsymbol{A}(\boldsymbol{x})=\boldsymbol{F}'(\boldsymbol{x})$，则得牛顿法的局部收敛性定理.

定理 9.3.3 设 $\boldsymbol{x}^*\in S_0\subset D$ 是方程(9.1.2)的解，$\boldsymbol{F}$: $D\subset\mathbb{R}^n\to\mathbb{R}^n$在$\boldsymbol{x}^*$的开邻域 S_0 上 F-可导，且 $\boldsymbol{F}'(\boldsymbol{x}^*)$非奇异，则存在 $S=\bar{S}(\boldsymbol{x}^*,\delta)\subset S_0(\delta>0)$，使映射 $\boldsymbol{G}(\boldsymbol{x})=\boldsymbol{x}-\boldsymbol{F}'(\boldsymbol{x})^{-1}\boldsymbol{F}(\boldsymbol{x})$对所有$\boldsymbol{x}\in S$ 有定义，且牛顿法(9.3.2)的迭代序列$\{\boldsymbol{x}^k\}$超线性收敛到 $\boldsymbol{x}^*$. 若再假定

$$\| \boldsymbol{F}'(\boldsymbol{x}) - \boldsymbol{F}'(\boldsymbol{x}^*) \| \leqslant \alpha \| \boldsymbol{x} - \boldsymbol{x}^* \| \quad \forall \boldsymbol{x}\in s,$$

其中 $\alpha>0$ 为常数，则牛顿法(9.3.2)至少二阶收敛.

如果在定理中还假定 $\boldsymbol{F}(\boldsymbol{x})$在 $\boldsymbol{x}^*$ 处二阶 F-可导且对任何 $\boldsymbol{h}\in\mathbb{R}^n$有$\boldsymbol{F}''(\boldsymbol{x}^*)\boldsymbol{h}\boldsymbol{h}\neq\boldsymbol{0}$，则牛顿法是二阶收敛的.

9.3.2 牛顿法的收敛性与误差估计

上节讨论的局部收敛性是在方程组(9.1.2)的解存在的前提下给出的，但通常情况并不知道方程组的解是否存在，本节给出的半局部收敛定理既给出了方程解在 $\boldsymbol{x}^0$ 的邻域内存在惟一，也给出牛顿法迭代序列$\{\boldsymbol{x}^k\}$收敛于解 $\boldsymbol{x}^*$ 及其误差估计，在这些定理中以康托洛维奇定理最为重要.

定理 9.3.4(康托洛维奇定理) 假定 $\boldsymbol{F}$: $D\subset\mathbb{R}^n\to\mathbb{R}^n$在凸集

从$(x_1^0, x_2^0)=(1,0)$出发，计算

$$F'(x^0)=\begin{pmatrix}8 & 0\\ 0.4597 & 1.5403\end{pmatrix},\quad F(x^0)=\begin{pmatrix}0\\ 0.159\end{pmatrix}.$$

由公式(9.3.3)算出 $x_1^1=1.0, x_2^1=-0.10292$，再由$(x_1^1, x_2^1)$出发，由公式(9.3.3)逐步迭代，表 9.1 给出了计算结果.

表 9.1　牛顿法计算例题

k	x_1^k	x_2^k	$f_1(x_1^k, x_2^k)$	$f_2(x_1^k, x_2^k)$
0	1.0	0.0	0.0	1.59E−1
1	1.0	−0.1029207	1.06E−2	4.55E−3
2	0.998609	−0.1055307	1.46E−5	6.63E−7
3	0.998607	−0.1055305	2.20E−11	−1.03E−11

上述构造牛顿法的方法实际上是用线性方程组(9.3.1)近似代替方程组(9.1.2)得到近似解，更一般地可用线性方程组

$$A(x^k)(x-x^k)=-F(x^k) \tag{9.3.4}$$

的解

$$x^{k+1}=x^k-[A(x^k)]^{-1}F(x^k)\quad (k=0,1,\cdots) \tag{9.3.5}$$

作为方程组(9.1.2)的近似解. 这种方法实际上就是线性化方法，这里 x^k 近似 x^*，且应要求 $A(x^k)\in\mathbb{R}^{n\times n}$ 近似于 $F'(x^*)$，基于不同考虑，适当选取 $A(x)$就可得到牛顿法的各种变型，称式(9.3.5)为牛顿型迭代法. 显然，取 $A(x)=F'(x)$，就是牛顿法；而取 $A(x)\equiv A\in\mathbb{R}^{n\times n}$就得到 n 维平行弦方法：

$$x^{k+1}=x^k-A^{-1}F(x^k)\quad (k=0,1,\cdots). \tag{9.3.6}$$

这是一维平行弦法(5.4.5)的推广，当 $A(x^k)\equiv F'(x^0)$时，就得到简化牛顿法：

$$x^{k+1}=x^k-F'(x^0)^{-1}F(x^k)\quad (k=0,1,\cdots). \tag{9.3.7}$$

对牛顿型迭代法(9.3.5)，有下面的局部收敛性定理.

定理 9.3.2　设 $x^*\in D$ 为方程组(9.1.2)的解，$F: D\subset\mathbb{R}^n\to$

$D_0 \subset D$ 上连续可微，且满足条件：

(1) $\| \boldsymbol{F}'(\boldsymbol{x})-\boldsymbol{F}'(\boldsymbol{y}) \| \leqslant \gamma \| \boldsymbol{x}-\boldsymbol{y} \|$, $\forall \boldsymbol{x},\boldsymbol{y} \in D_0$.

(2) $\| \boldsymbol{F}'(\boldsymbol{x}^0)^{-1}\boldsymbol{F}(\boldsymbol{x}^0) \| \leqslant \eta, \boldsymbol{x}^0 \in D_0$.

(3) $\| \boldsymbol{F}'(\boldsymbol{x}^0)^{-1} \| \leqslant \beta$.

(4) 令 $h=\beta\gamma\eta \leqslant \frac{1}{2}, t^* = \frac{\eta}{h}[1-\sqrt{1-2h}], t^{**} = \frac{\eta}{h}[1+\sqrt{1-2h}]$.

$$\bar{S}(\boldsymbol{x}^0,t^*) = \{\boldsymbol{x} \in \mathbb{R}^n \mid \| \boldsymbol{x}-\boldsymbol{x}^* \| \leqslant t^*\} \subset D.$$

则牛顿法(9.3.2)生成序列$\{\boldsymbol{x}^k\} \subset \bar{S}(\boldsymbol{x}^0,t^*)$，且收敛到方程 $\boldsymbol{F}(\boldsymbol{x})=0$ 的一个解 $\boldsymbol{x}^*$，在域 $S(\boldsymbol{x}^0,t^{**}) \cap D_0$ 内解 $\boldsymbol{x}^*$ 是惟一的，其误差估计

$$\| \boldsymbol{x}^* - \boldsymbol{x}^k \| \leqslant \frac{\eta}{h} \frac{(1-\sqrt{1-2h})^{2^k}}{2^k} \quad (k=0,1,\cdots). \tag{9.3.10}$$

注意，当 $h<\frac{1}{2}$时可得到牛顿法二阶收敛的结论，而当 $h=\frac{1}{2}$时，由(9.3.10)得到

$$\| \boldsymbol{x}^* - \boldsymbol{x}^k \| \leqslant \frac{1}{2^{k-1}}\eta,$$

此时只有线性收敛，关于牛顿法的半局部收敛性较重要的还有如下的梅索夫斯基定理.

定理 9.3.5 假定 $\boldsymbol{F}: D \subset \mathbb{R}^n \to \mathbb{R}^n$在凸集 $D_0 \subset D$ 上 F-可导，且对任何 $\boldsymbol{x} \in D_0$ 有 $\| \boldsymbol{F}'(\boldsymbol{x})^{-1} \| \leqslant \beta$，对 $\forall \boldsymbol{x},\boldsymbol{y} \in D_0$，

$$\| \boldsymbol{F}'(\boldsymbol{x}) - \boldsymbol{F}'(\boldsymbol{y}) \| \leqslant \gamma \| \boldsymbol{x}-\boldsymbol{y} \|,$$

若 $\boldsymbol{x}^0 \in D_0$ 使 $\| \boldsymbol{F}'(\boldsymbol{x}^0)^{-1}\boldsymbol{F}(\boldsymbol{x}^0) \| \leqslant \eta$，且 $\alpha = \frac{1}{2}\beta\gamma\eta < 1$，记 $\gamma_0 = \eta\sum_{j=1}^{\infty}\alpha^{2j-1}$，若闭球 $S=\bar{S}(\boldsymbol{x}^0,\gamma_0) \subset D_0$，则牛顿法(9.3.2)生成的序

列$\{\boldsymbol{x}^k\}\subset S$并收敛于方程(9.1.2)的一个解$\boldsymbol{x}^*\in S$. 此外

$$\|\boldsymbol{x}^*-\boldsymbol{x}^k\|\leqslant \varepsilon_k\|\boldsymbol{x}^k-\boldsymbol{x}^{k-1}\|^2\quad(k=1,2,\cdots),\tag{9.3.11}$$

其中$\varepsilon_k=\dfrac{\alpha}{\eta}\sum_{j=0}^{\infty}(\alpha^{2^k})^{2j-1}\leqslant \alpha[\eta(1-\alpha^{2^k})]^{-1}$.

牛顿法具有收敛快的优点，在计算时每一步计算只与前一步有关，误差不传播，是自校正的，因此，在理论和实际应用上都是一种重要方法. 但是牛顿法对初始近似$\boldsymbol{x}^0$限制较大，只有$\boldsymbol{x}^0$与解$\boldsymbol{x}^*$靠近才能保证收敛. 另外，每步要算n^2个分量偏导数值和n个分量函数值，还要解一个线性方程组，故工作量较大，针对这些缺点牛顿法有不少改进的算法.

9.3.3 离散牛顿法与修正牛顿法

在实际使用牛顿法时，为了避免计算$\boldsymbol{F}'(\boldsymbol{x})$，通常可用差商代替偏导数，即用矩阵$\boldsymbol{J}(\boldsymbol{x},\boldsymbol{h})$代替$\boldsymbol{F}'(\boldsymbol{x})$.

$$\boldsymbol{J}(\boldsymbol{x},\boldsymbol{h})=\begin{pmatrix}\dfrac{1}{h_1}[f_1(\boldsymbol{x}+h_1\boldsymbol{e}_1)-f_1(\boldsymbol{x})] & \cdots & \dfrac{1}{h_n}[f_1(\boldsymbol{x}+h_n\boldsymbol{e}_n)-f_1(\boldsymbol{x})]\\ \vdots & & \vdots\\ \dfrac{1}{h_1}[f_n(\boldsymbol{x}+h_1\boldsymbol{e}_1)-f_n(\boldsymbol{x})] & \cdots & \dfrac{1}{h_n}[f_n(\boldsymbol{x}+h_n\boldsymbol{e}_n)-f_n(\boldsymbol{x})]\end{pmatrix},\tag{9.3.12}$$

这里$\boldsymbol{h}=(h_1,\cdots,h_n)^{\mathrm{T}}$，$\boldsymbol{e}_i$为第$i$个坐标向量. 如果$\boldsymbol{J}(\boldsymbol{x},\boldsymbol{h})^{-1}$存在，用$\boldsymbol{J}(\boldsymbol{x}^k,\boldsymbol{h}^k)^{-1}$代替式(9.3.2)的$\boldsymbol{F}'(\boldsymbol{x}^k)^{-1}$，则得**离散牛顿法**：

$$\boldsymbol{x}^{k+1}=\boldsymbol{x}^k-\boldsymbol{J}(\boldsymbol{x}^k,\boldsymbol{h}^k)^{-1}\boldsymbol{F}(\boldsymbol{x}^k)\quad(k=0,1,\cdots),\tag{9.3.13}$$

其中$\{\boldsymbol{h}^k\}$为预先给定的向量序列. 若取

$$\boldsymbol{h}^k=(f_1(\boldsymbol{x}^k),f_2(\boldsymbol{x}^k),\cdots,f_n(\boldsymbol{x}^k))^{\mathrm{T}}=\boldsymbol{F}(\boldsymbol{x}^k),$$

这时$\boldsymbol{J}(\boldsymbol{x}^k,\boldsymbol{h}^k)=\boldsymbol{J}(\boldsymbol{x}^k,\boldsymbol{F}(\boldsymbol{x}^k))$，将它代入式(9.3.13)则得

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \boldsymbol{J}(\boldsymbol{x}^k, \boldsymbol{F}(\boldsymbol{x}^k))^{-1}\boldsymbol{F}(\boldsymbol{x}^k) \quad (k = 0,1,\cdots). \tag{9.3.14}$$

此即**牛顿-斯蒂芬森方法**.这个算法同样具有平方敛速,每步计算$(n+1)$个$\boldsymbol{F}(\boldsymbol{x})$函数值,如果计算工作量$W$以每步计算函数值的个数衡量,则$W=n+1$,而收敛阶$p=2$,于是由式(9.2.4)可知此算法效率

$$e_N = \frac{\ln 2}{n+1}. \tag{9.3.15}$$

在牛顿法中,如果把计算$\boldsymbol{F}'(\boldsymbol{x}^k)$与计算$n$个$\boldsymbol{F}(\boldsymbol{x}^k)$等同,则其效率也为$e_N$.程序(9.3.14)仅把计算偏导数变成计算函数值,以便于在计算机上使用,其余的效果与牛顿法相同.

为了减少牛顿法的计算量,通常可用简化牛顿法,即

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \boldsymbol{F}'(\boldsymbol{x}^0)^{-1}\boldsymbol{F}(\boldsymbol{x}^k) \quad (k = 0,1,\cdots). \tag{9.3.16}$$

这种方法除开始计算$\boldsymbol{F}'(\boldsymbol{x}^0)^{-1}$外,以后每步只计算一个函数值,减少计算量,但这个程序只有线性敛速,收敛慢,故算法(9.3.16)效率仍很低.沙曼斯基(Шаманский)把m步简化牛顿法组成一步牛顿法,得到下列程序:

$$\begin{cases} \boldsymbol{x}^{k,0} = \boldsymbol{x}^0, \\ \boldsymbol{x}^{k,i} = \boldsymbol{x}^{k,i-1} - \boldsymbol{F}'(\boldsymbol{x}^k)^{-1}\boldsymbol{F}(\boldsymbol{x}^{k,i-1}), \\ \boldsymbol{x}^{k+1} = \boldsymbol{x}^{k,m}, \end{cases} \tag{9.3.17}$$
$$i = 1,\cdots,m, \quad k = 0,1,\cdots$$

称为**修正牛顿法**.这个程序由$\boldsymbol{x}^k$计算到$\boldsymbol{x}^{k+1}$只计算一次$\boldsymbol{F}'(\boldsymbol{x}^k)$,相当于计算$n$个$\boldsymbol{F}$值,求一次$\boldsymbol{F}'(\boldsymbol{x}^k)^{-1}$,和$m$个$\boldsymbol{F}$的函数值,总共计算函数值的次数为$n+m$,即$w=n+m$.程序(9.3.17)生成的序列具有$m+1$阶敛速,即$p=m+1$.这个算法的效率

$$e_{N_m} = \frac{\ln(m+1)}{n+m}.$$

当$m=1$时,此方法即是牛顿法.$e_{N_1}=e_N$,由此又有

$$\frac{e_{N_m}}{e_N} = \frac{n+1}{n+m} \cdot \frac{\ln(m+1)}{\ln 2} \quad (m \leqslant n+1), \quad (9.3.18)$$

当 $m \geqslant 2$ 时显然有 $\frac{e_{N_m}}{e_N} > 1$，这说明算法(9.3.17)的效率高于牛顿法. 对给出的 n，如果可以选最优的 m，使式(9.3.18)取最大值，则有相应的结果，见表 9.2.

表 9.2 修正牛顿法与牛顿法的效率比

n	2	3	5	10	50	100	1000
m	3	3	5	7	22	37	225
$\frac{e_{N_m}}{e_{N_1}}$	1.20	1.33	1.55	1.94	3.20	3.87	6.39

迭代法(9.3.17)比牛顿法效率高，又包含了牛顿法，因此，它比牛顿法更有效，是一个可供实际使用的算法. 计算时，$\boldsymbol{F}'(\boldsymbol{x}^k)$也可用 $\boldsymbol{J}(\boldsymbol{x}^k, \boldsymbol{F}(\boldsymbol{x}^k))$代替.

9.3.4 牛顿下山法

为了扩大牛顿法的收敛范围，通常可构造**牛顿下山程序**

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \omega_k \boldsymbol{F}'(\boldsymbol{x}^k)^{-1} \boldsymbol{F}(\boldsymbol{x}^k) \quad (k = 0, 1, \cdots), \tag{9.3.19}$$

其中 ω_k 称为**下山参数**，$0 < \omega_k \leqslant 1$，可选择 ω_k 使

$$\| \boldsymbol{F}(\boldsymbol{x}^{k+1}) \| < \| \boldsymbol{F}(\boldsymbol{x}^k) \| \tag{9.3.20}$$

成立. 由于牛顿法(9.3.2)没有保证下山条件(9.3.20)成立，所以引入参数 ω_k 以后，就可使(9.3.20)的条件成立，从而使迭代序列收敛.

定理 9.3.6 设 $\boldsymbol{F}$: $D \subset \mathbb{R}^n \to \mathbb{R}^n$ 连续可微，$\| \boldsymbol{F}(\boldsymbol{x}^0) \| \leqslant \eta$，在域 $D_0 = \{\boldsymbol{x} \mid \| \boldsymbol{x} - \boldsymbol{x}^0 \| \leqslant \rho\beta\eta\} \subset D$ 上 $\boldsymbol{F}'(\boldsymbol{x})$可逆，且

$$\| \boldsymbol{F}'(\boldsymbol{x})^{-1} \| \leqslant \beta,$$

此外

$$\| \boldsymbol{F}'(\boldsymbol{x}) - \boldsymbol{F}'(\boldsymbol{y}) \| \leqslant \gamma \| \boldsymbol{x} - \boldsymbol{y} \|, \quad \forall \boldsymbol{x}, \boldsymbol{y} \in D_0$$

成立. 令 $\alpha = \gamma\beta^2\eta$, 则当 $\rho \geqslant \frac{2\alpha}{\epsilon^2} > 1$ 时方程(9.1.1)在 D_0 中有解 $\boldsymbol{x}^*$ 存在, 且在 $0 < \omega_k \leqslant 1$ 及 $0 < \epsilon \leqslant w_k\alpha \leqslant 2 - \epsilon$ 时, 由式(9.3.19)生成的序列 $\{\boldsymbol{x}^k\}$ 至少线性收敛于 $\boldsymbol{x}^*$.

由定理可知, 当给定 $\boldsymbol{x}^0 \in D$ 后, 每步迭代总可找到满足定理要求的参数 $\omega_k \leqslant 1$, 使迭代法(9.3.19)收敛, 在这个意义下牛顿下山法是大范围收敛的. 但在实际计算时, 要选择满足定理条件的 ω_k 仍较困难, 因此, 可以先令 $\omega_k = 1$, 用逐次减半方法确定 ω_k, 只要检验条件(9.3.20)成立即可. 这样做当然增加了计算量, 但却减少了对初始近似 $\boldsymbol{x}^0$ 的限制. 实际计算时还可用 $\boldsymbol{J}(\boldsymbol{x}^k, \boldsymbol{F}(\boldsymbol{x}^k))^{-1}$ 代替 $\boldsymbol{F}'(\boldsymbol{x}^k)^{-1}$, 从而使式(9.3.19)改为

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \omega_k \boldsymbol{J}(\boldsymbol{x}^k, \boldsymbol{F}(\boldsymbol{x}^k))^{-1} \boldsymbol{F}(\boldsymbol{x}^k) \quad (k = 0, 1, \cdots) \tag{9.3.21}$$

9.4 布朗方法与布兰特方法

9.4.1 布朗方法

在牛顿法(9.3.2)中, 把非线性映射 $\boldsymbol{F}$ 作为一个整体逐次线性化, 然后求解牛顿方程组(9.3.1)得到 $\Delta\boldsymbol{x}^k$. 这种方法便于进行理论分析, 又能简化迭代格式表示, 但未充分利用函数 $\boldsymbol{F}$ 的具体结构. 当各分量函数 $f_i (i = 1, 2, \cdots, n)$ 非线性程度分布不均时, 把所有分量作为一个整体考虑显然对计算效率的提高是不利的, 为此, 有必要按分量方程 $f_i(x) = 0 (i = 1, 2, \cdots, n)$ 逐个加以考虑. 布朗(Brown)正是基于这种考虑, 于 1966 年首先提出了对牛顿法的改进, 布朗算法的基本思想是对 $\boldsymbol{F}$ 各个分量 f_i 逐个线性化, 并用其中每一个线性方程消去余下那些非线性方程的一个变量, 得到

一个上三角形的线性方程组，再逐个回代求得新近似 $\boldsymbol{x}^{k+1}$ 的各分量. 这样做减少了计算 $\boldsymbol{F}$ 函数值的个数，又具有平方敛速，从而提高了效率. 布朗方法很大程度依赖于 $\boldsymbol{F}$ 和 $\boldsymbol{x}$ 的分量顺序，为了方便，叙述是以原分量次序顺序进行，但使用时则应按一定原则排列次序. 设方程(9.1.2)的 k 次近似解为 $\boldsymbol{x}^k=(x_1^k,x_2^k,\cdots,x_n^k)$，由 $\boldsymbol{x}^k$ 计算 $\boldsymbol{x}^{k+1}$ 的步骤如下：

第 1 步：对方程 $F(x)=0$ 的第一个方程 $f_1(x)=0$ 用线性方程

$$f_1(\boldsymbol{x}^k)+\sum_{j=1}^{n}\Delta_{1j}^k(x_j-x_j^k)=0 \tag{9.4.1}$$

近似代替，其中

$$\Delta_{1j}^k=\frac{1}{h_j^k}[f_1(\boldsymbol{x}^k+h_j^k\boldsymbol{e}_j)-f_1(\boldsymbol{x}^k)]$$

$$\approx\frac{\partial f_1(\boldsymbol{x}^k)}{\partial x_j}\quad(j=1,2,\cdots,n),$$

这里实际上是用 $f_1(\boldsymbol{x})$ 在点 $\boldsymbol{x}^k$ 的一阶泰勒近似逼近 $f_1(\boldsymbol{x})$，但为了避免计算偏导数 $\frac{\partial f_1(\boldsymbol{x}^k)}{\partial x_j}$ 而用差商代替. 假定 $\Delta_{11}^k\neq0$，且在 $\Delta_{1j}^k(j=1,2,\cdots,n)$ 中绝对值最大，则由方程(9.4.1)可求得

$$x_1=x_1^k-(\Delta_{11}^k)^{-1}[f_1(\boldsymbol{x}^k)+\sum_{j=2}^{n}\Delta_{1j}^k(x_j-x_j^k)]=L_1(x_2,x_3,\cdots,x_n). \tag{9.4.2}$$

在实际计算时，应选 Δ_{1j} 中绝对值最大者，求出 x_j，即选主元.

第 2 步：将式(9.4.2)的 x_1 代入方程(9.1.1)其余各方程，并记

$$g_2(x_2,x_3,\cdots,x_n)=f_2(L_1(x_2,x_3,\cdots,x_n),x_2,x_3,\cdots,x_n),$$

$$g_2^k=f_2(L_1^k,x_2^k,\cdots,x_n^k),L_1^k=L_1(x_2^k,x_3^k,\cdots,x_n^k),$$

用与第 1 步同样的方法，对 $g_2(x_2,x_3,\cdots,x_n)$ 用 $\boldsymbol{x}^k$ 处一阶泰勒近

似，并解出 x_2，得到

$$x_2 = x_2^k - (\Delta_{22}^k)^{-1}\left[g_2^k + \sum_{j=3}^{n} \Delta_{2j}^k (x_j - x_i^k)\right] = L_2(x_3, x_4, \cdots, x_n),$$

其中

$$\Delta_{2j}^k = \frac{1}{h_j^k}[g_2(x_2^k, x_3^k, \cdots, x_j^k + h_j^k e_j, \cdots, x_n^k) - g_2^k] \quad (j = 2, 3, \cdots, n).$$

第 i 步：将 $x_1 = L_1(x_2, x_3, \cdots, x_n), \cdots, x_{i-1} = L_{i-1}(x_i, x_{i+1}, \cdots, x_n)$ 代入其余 $n-i+1$ 个方程，记

$$g_i(x_i, x_{i+1}, \cdots, x_n) = f_i(L_1, \cdots, L_{i-1}, x_i, \cdots, x_n),$$

其中 L_i 是由 $i-1$ 列三角线性方程组回代得到，所有 L_i 都要用 $x_i, \cdots, x_n$ 表示，用与前面步骤相同的过程可求得

$$\begin{aligned} x_i &= x_i^k - (\Delta_{ii}^k)^{-1}\left[g_i^k + \sum_{j=i+1}^{n} \Delta_{ij}^k (x_j - x_j^k)\right] \\ &= L_i(x_{i+1}, \cdots, x_n), \end{aligned} \tag{9.4.3}$$

其中

$$\begin{aligned} \Delta_{ij}^k = \frac{1}{h_j^k}[g_i(x_i^k, \cdots, x_{j-1}^k, x_j^k + \\ h_j^k e_j, x_{j+1}^k, \cdots, x_n^k) - g_i^k] \quad (j = i, \cdots, n). \end{aligned} \tag{9.4.4}$$

这样每步消去一个变量.

第 n 步：由公式(9.4.3)，当 i 从 1 到 n 时有

$$g_n(x_n) = f_n(L_1, \cdots, L_{n-1}, x_n),$$

其中 $L_i(i=1, \cdots, n-1)$ 都是 x_n 的函数，由此求出

$$x_n = x_n^k - (\Delta_{nn}^k)^{-1} g_n^k = L_n.$$

把它作为 $\boldsymbol{x}^*$ 的第 n 个分量 x_n^* 的第 $k+1$ 次近似，记作 x_n^{k+1}，即

$$x_n^{k+1} = x_n^k - (\Delta_{nn}^k)^{-1} g_n^k,$$

其中 Δ_{nn}^k 是 $g_n(x_n)$ 在点 x_n^k 处的导数近似. 将它代入式(9.4.3)，这就是“回代过程”，可逐步得到

$$x_i^{k+1} = L_i(x_{i+1}^{k+1},\cdots,x_n^{k+1}) \quad (i = n-1,\cdots,2,1). \qquad (9.4.5)$$

按以上步骤从 $\boldsymbol{x}^0$ 开始反复迭代，直到 $\boldsymbol{x}^m$ 满足精度要求为止. 考虑到步长与函数值、变量值及机器字长等的关系，对第 i 个分量函数 g_i，h_j^k 按如下规定选取：

$$h_j^k = \max\{\alpha_{ij}^k, 5\times 10^{-\beta+2}\} \quad (j = 1,2,\cdots,n),$$

而

$$\alpha_{ij}^k = \min\{\max(|f_1^{(k)}|,|g_2^k|,\cdots,|g_i^k|), 0.01\boldsymbol{x}|x_j^k|\},$$

其中 β 为机器的有效数位(十进制).

根据以上算法可知，布朗算法每迭代一步所需计算函数值个数是：$i=1$ 时要算 f_1 的 $n+1$ 个值，$i=2$ 时要算 f_2 的 n 个值，以此类推，由 $\boldsymbol{x}^k$ 到 $\boldsymbol{x}^{k+1}$ 迭代一步需要计算

$$\sum_{i=2}^{n+1} i = \frac{n^2+3n}{2}$$

个分量函数值，相当于 $\frac{n+3}{2}$ 个 $\boldsymbol{F}$ 值，比牛顿法减少将近一半计算量. 布朗算法生成的序列 $\{\boldsymbol{x}^k\}$ 是平方收敛的，它的效率

$$e_{B_1} = \frac{2\ln 2}{n+3}.$$

因为布朗方法每次只对一个分量方程 $f_i=0$ 进行运算，所以就存在一个最优次序问题，使用时应先把线性方程排在前面，然后按非线性程度由低次到高次排列，这也是布朗算法的一个优点.

例 9.4.1 设非线性方程组为

$$\begin{cases} x_1^2 + x_2^2 - x_3 - 2 = 0, \\ x_1 + 5x_2 + 1 = 0, \\ x_1x_3 - 2x_1 + 1 = 0. \end{cases}$$

用布朗方法求 $\boldsymbol{x}^0 = (-2,0,2)$ 附近的根.

解 先将给定方程组按线性情况排列，令

$$\begin{cases} f_1(x) = x_1 + 5x_2 + 1 = 0, \\ f_2(x) = x_1 x_3 - 2x_1 + 1 = 0, \\ f_3(x) = x_1^2 + x_2^2 - x_3 - 2 = 0. \end{cases}$$

因 f_1 已是线性，按主元消去 x_2，则得

$$x_2 = -\frac{1}{5}(x_1 + 1) = L_1(x_1, x_3),$$

代入 f_2, f_3，对 f_2 用线性函数近似，即用

$$l_2(\boldsymbol{x}) = f_2^k + (x_3^k - 2)(x_1 - x_1^k) + x_1^k(x_3 - x_3^k) = 0$$

近似 $f_2(x)=0$，解出

$$x_3 = x_3^k - \frac{1}{x_1^k}[(x_3^k - 2)x_1 + 1] = L_2(x_1).$$

将它代入 f_3，得

$$g_3(x_1) = x_1^2 + \frac{1}{25}(x_1 + 1)^2 - \left\{x_3^k - \frac{1}{x_1^k}[(x_3^k - 2)x_1 + 1]\right\} - 2,$$

对 $g_3(x_1)=0$ 用一步牛顿法，得

$$x_1^{k+1} = x_1^k - \frac{(x_1^k)^2 + \frac{1}{25}(x_1^k + 1)^2 - \left\{x_3^k - \frac{1}{x_1^k}[(x_3^k - 2)x_1^k + 1]\right\} - 2}{\frac{52}{25}x_1^k + \frac{2}{25} + \frac{x_3^k - 2}{x_1^k}},$$

$$k = 0, 1, \cdots \tag{9.4.6}$$

将得到的 x_1^{k+1} 进行回代，得到

$$\begin{cases} x_2^{k+1} = -\frac{1}{5}(x_1^{k+1} + 1), \\ x_3^{k+1} = x_3^k - \frac{1}{x_1^k}[(x_3^k - 2)x_1^{k+1} + 1], \end{cases} \quad k = 0, 1, \cdots \tag{9.4.7}$$

用式(9.4.6)和式(9.4.7)的迭代过程，从 $\boldsymbol{x}^0$ 出发可得到下列计算结果：

k	x_1^k	x_2^k	x_3^k
0	−2.000000	0.000000	2.000000
1	−2.112747	0.222549	2.500000
2	−2.103978	0.220796	2.475393
3	−2.103937	0.220787	2.475299
4	−2.103937	0.220787	2.475299

在例 9.4.1 中，Δ_{ij} 均用偏导数$\dfrac{\partial f_i}{\partial x_j}$，而不用差商近似，但在计算机上为避免计算偏导数，通常采用差商近似 Δ_{ij}.

9.4.2 布兰特方法

布兰特(Brent)于 1973 年在布朗方法基础上使用沙曼斯基技巧，即在由 $\boldsymbol{x}^k$ 计算 $\boldsymbol{x}^{k+1}$时，用了 m 步布朗方法，且每步用的 Δ_{ij}^k 不变. 这时，计算回代过程的公式可表示为

$$\begin{cases} x_i^{k,l+1} = x_i^{k,l} - (\Delta_{ij}^k)^{-1}\left[g_i(x^{k,l}) + \sum\limits_{j=i+1}^{n} \Delta_{ki}^j (x_k^{j,l+1} - x_l^{k,l})\right] \\ x_n^{k,l+1} = x_n^{k,l} - (\Delta_{nn}^k)^{-1} g_n^{kl} \\ \qquad\qquad i = n-1,\cdots,1, \quad l = 0,1,\cdots,m-1, \end{cases} \tag{9.4.8}$$

其中 $\boldsymbol{x}^{k,0} = \boldsymbol{x}^k$，$\boldsymbol{x}^{k,m} = \boldsymbol{x}^{k+1}$，$\Delta_{ij}^k$ 由式(9.4.4)表示. 这就是布兰特方法的基本思想. 这时由 $\boldsymbol{x}^k$ 计算 $\boldsymbol{x}^{k+1}$ 的工作量比一步布朗法大，多算了 $(m-1)$ 个函数值. 故总工作量为 $w = \dfrac{n+3}{2} + m - 1 = \dfrac{n+2m+1}{2}$次函数值.

然而，布兰特方法在具体实现时并不采用布朗方法的消去与回代算法，而使用一种基于正交变换的方法，即假定 k 次迭代后，已有 $\boldsymbol{x}^*$ 的 k 次近似 $\boldsymbol{x}^k$ 及一个正交矩阵 $\boldsymbol{Q}_k$ 和一个步长 $h_k > 0$（初始 $\boldsymbol{x}^0$，$h_0 > 0$，给定 $\boldsymbol{Q}_0 = \boldsymbol{I}$），则布兰特方法生成 $\boldsymbol{x}^{k+1}$，h_{k+1}，及 $\boldsymbol{Q}_{k+1}$

的计算步骤如下：

1. 令 $\boldsymbol{Q}_{k,1}=\boldsymbol{Q}_k,\boldsymbol{y}^{k,1,0}=\boldsymbol{x}^k$；

2. 对 $j=1,2,\cdots,n$ 做步骤 3～5；

3. 计算

$$\boldsymbol{a}_{kj}=\frac{1}{h_k}\begin{pmatrix}0\\ \vdots\\ 0\\ f_i(\boldsymbol{y}^{k,j,0}+h_k\boldsymbol{Q}_{kj}\boldsymbol{e}_j)-f_i(\boldsymbol{y}^{k,j,0})\\ \vdots\\ f_j(\boldsymbol{y}^{k,j,0}+h_k\boldsymbol{Q}_{kj}\boldsymbol{e}_n)-f_j(\boldsymbol{y}^{k,j,0})\end{pmatrix}$$

4. 令 $\sigma_{kj}=\pm\|\boldsymbol{a}_{kj}\|$，找一个正交矩阵 $\boldsymbol{U}_{kj}$，形如

$$\boldsymbol{U}_{kj}=\begin{pmatrix}\boldsymbol{I}_{j-1} & \boldsymbol{0}\\ \boldsymbol{0} & \hat{\boldsymbol{U}}_{kj}\end{pmatrix}\begin{matrix}\}j-1\\ \}n-j+1\end{matrix}$$

使 $\boldsymbol{U}_{kj}^{\mathrm{T}}\boldsymbol{a}_{kj}=\sigma_{kj}\boldsymbol{e}_j$，这里 $\hat{\boldsymbol{U}}_{kj}$ 取为初等反射阵，于是 $\boldsymbol{U}_{kj}$ 可表示为

$$\boldsymbol{U}_{kj}=\boldsymbol{I}-2\boldsymbol{w}_j\boldsymbol{w}_j^{\mathrm{T}},$$

其中 $\boldsymbol{w}_j\in\mathbb{R}^n$，具有形式

$$\boldsymbol{w}_j^{\mathrm{T}}=(0,\cdots,0,w_{jj},\cdots,w_{jn}),$$

且 $\boldsymbol{w}_j^{\mathrm{T}}\boldsymbol{w}_j=1$.

5. 计算

$$\boldsymbol{Q}_{k,j+1}=\boldsymbol{Q}_{kj}\boldsymbol{U}_{kj}$$

及

$$\boldsymbol{y}^{k,j+1,0}=\boldsymbol{y}^{k,j,0}-\left(\frac{f_j(\boldsymbol{y}^{k,j,0})}{\sigma_{kj}}\right)\boldsymbol{Q}_{k,j+1}\boldsymbol{e}_j.$$

6. 对 $l=1,\cdots,m$，令 $\boldsymbol{y}^{k,1,l}=\boldsymbol{y}^{k,n+1,l-1}$，对 $j=1,\cdots,n$ 计算

$$\boldsymbol{y}^{k,j+1,l}=\boldsymbol{y}^{k,j,l}-\left(\frac{f_j(\boldsymbol{y}^{k,j,l})}{\sigma_{kj}}\right)\boldsymbol{Q}_{k,n+1}\boldsymbol{e}_j.$$

7. 令 $\boldsymbol{x}^{k+1}=\boldsymbol{y}^{k,n+1,m}$，计算 $\boldsymbol{F}(\boldsymbol{x}^{k+1})$，若 $\|\boldsymbol{F}(\boldsymbol{x}^{k+1})\|\leqslant\varepsilon_2$ 或 $\|\boldsymbol{x}^{k+1}-\boldsymbol{x}^k\|\leqslant\varepsilon_1\|\boldsymbol{x}^{k+1}\|$（其中 $\varepsilon_1,\varepsilon_2$ 为给定精度要求），则置 $\boldsymbol{x}^*=\boldsymbol{x}^{k+1}$，打印 $\boldsymbol{x}^*$ 后结束；否则 $k+1\to k,\boldsymbol{x}^{k+1}\to\boldsymbol{x}^k,\boldsymbol{Q}_{k+1}=\boldsymbol{Q}_{k,n+1}\to\boldsymbol{Q}_k$

此即为牛顿方程组..对有的问题(如大型,稀疏方程)用迭代法求解方程组(9.5.1)比直接法更方便.如用迭代法解,则在牛顿法每步中又生成一个解线性方程组的迭代序列$\{x^{k,j}, j=0,1,\cdots\}$,这就形成一个双层迭代,其中$k=0,1,\cdots$表示牛顿迭代,对每个$k$又有$j=0,1,\cdots$表示的线性方程组迭代做辅助迭代.从理论上说,一切求解线性方程组的迭代法都可用于求方程(9.5.1)的解,其中,用SOR迭代求解牛顿方程组被称为**牛顿-SOR迭代**,简记**N-SOR迭代.**

此外,把解线性方程组的SOR迭代直接用于非线性方程组(9.1.2),在每松弛步中解一个单变量非线性方程,如用牛顿法求此方程的解,则可得到一种以SOR为主、以牛顿法为辅助迭代的**SOR-牛顿迭代法.**

9.5.1 牛顿-SOR迭代法

若记$\boldsymbol{A}_k=\boldsymbol{F}'(\boldsymbol{x}^k)$,$\boldsymbol{b}_k=\boldsymbol{F}'(\boldsymbol{x}^k)\boldsymbol{x}^k-\boldsymbol{F}(\boldsymbol{x}^k)$,则方程(9.5.1)可改写成

$$\boldsymbol{A}_k\boldsymbol{x}^{k+1}=\boldsymbol{b}_k \quad (k=0,1,\cdots). \tag{9.5.2}$$

将$\boldsymbol{A}_k$分裂为$\boldsymbol{A}_k=\boldsymbol{B}_k-\boldsymbol{C}_k$,其中$\boldsymbol{B}_k$非奇,且$\boldsymbol{B}_k^{-1}$容易计算.例如$\boldsymbol{B}_k$为对角阵、下三角阵时,可构造迭代法:

$$\boldsymbol{x}^{k,j+1}=\boldsymbol{B}_k^{-1}\boldsymbol{C}_k\boldsymbol{x}^{k,j}+\boldsymbol{B}_k^{-1}\boldsymbol{b}_k \quad (j=0,1,\cdots).$$

记$\boldsymbol{H}_k=\boldsymbol{B}_k^{-1}\boldsymbol{C}_k$,则上式可写成

$$\boldsymbol{x}^{k,j+1}=\boldsymbol{H}_k\boldsymbol{x}^{k,j}+\boldsymbol{B}_k^{-1}\boldsymbol{b}_k \quad (j=0,1,\cdots).$$

若$\lim\limits_{j\to\infty}\boldsymbol{x}^{k,j}$存在,在迭代$j_k$次后,如果

$$\|\boldsymbol{x}^{k,j_k}-\boldsymbol{x}^{k+1}\|\leqslant\varepsilon,$$

则有$\boldsymbol{x}^{k+1}\approx\boldsymbol{x}^{k,j_k}$,这里$\boldsymbol{x}^{k+1}$表示方程(9.5.2)的精确解.如果直接取

$$\begin{aligned}\boldsymbol{x}^{k+1}=\boldsymbol{x}^{k,j_k}&=\boldsymbol{H}_k\boldsymbol{x}^{k,j_k-1}+\boldsymbol{B}_k^{-1}\boldsymbol{b}_k\\&=\boldsymbol{H}_k(\boldsymbol{H}_k\boldsymbol{x}^{k,j_k-2}+\boldsymbol{B}_k^{-1}\boldsymbol{b}_k)+\boldsymbol{B}_k^{-1}\boldsymbol{b}\boldsymbol{x}\end{aligned}$$

$$h_{k+1}=\begin{cases}-f_1(x^{k+1})/\sigma_{k,1}, & 当\ f_1(x^{k+1})\neq 0,\\ \epsilon\|\boldsymbol{x}^{k+1}\|, & 当\ f_1(x^{k+1})=0,\end{cases}$$

$h_{k+1}\to h_k$，转步骤 1.

定理 9.4.2 假定 $\epsilon>0, n>1, S=\{\boldsymbol{x}\in\mathbb{R}^n \mid \|\boldsymbol{x}-\boldsymbol{x}^*\|<3\epsilon\}$，$\boldsymbol{F}: S\to\mathbb{R}^n$ 是 F-可微的，且 $\boldsymbol{F}(\boldsymbol{x}^*)=\boldsymbol{0}$. 若 $\boldsymbol{F}'(\boldsymbol{x}^*)$ 非奇异，$h_0<\epsilon$ 且 $\|\boldsymbol{x}^0-\boldsymbol{x}^*\|<\epsilon$，当 ϵ 充分小，由布兰特方法生成的序列 $\{\boldsymbol{x}^k\}$ 收敛于 $\boldsymbol{x}^*$，且收敛阶至少为 $m+1$.

根据式(9.2.4)，布兰特方法的效率

$$e(B_m)=\frac{2\ln(m+1)}{n+2m+1}, \tag{9.4.9}$$

当 $m=1$ 时，

$$e(B_1)=e_{B_1}=\frac{2\ln 2}{n+3}.$$

在式(9.4.9)中当 n 给定时，可求出最优的 m，记作 $\bar{m}$，即 $e(B_{\bar{m}})=\max\limits_m e(B_m)$，$\bar{m}$ 值见表 9.3.

表 9.3 布朗法与布兰特法的效率比

n	2	3	5	8	10	20	30	50	100	1000
$\bar{m}$	2	3	3	4	5	7	10	14	22	128
$e(B_1)/e(N_1)$	1.20	1.33	1.50	1.64	1.69	1.83	1.88	1.92	1.96	2.00
$e(B_{\bar{m}})/e(N_1)$	1.36	1.60	2.00	2.46	2.71	3.60	4.21	5.04	6.30	11.17

布兰特方法取最优 m 值时效率最高，因此，目前不少计算机的数学库中都有用布兰特方法编制的软件.

9.5 牛顿松弛型迭代法

在牛顿法中，每步迭代都要解一个线性方程组

$$\boldsymbol{F}'(\boldsymbol{x}^k)\Delta\boldsymbol{x}^k=-\boldsymbol{F}(\boldsymbol{x}^k)\quad(k=0,1,\cdots), \tag{9.5.1}$$

(Newton-Seidel)**迭代法**. 在 N-SOR 迭代式(9.5.6)中，松弛因子 ω_k 及迭代次数 j_k 如何取，对收敛速度及计算效率均有较大影响，从计算效率考虑要求选最佳松弛因子 ω_k，使 N-SOR 迭代式(9.5.6)生成的迭代序列$\{\boldsymbol{x}^k\}$收敛最快. 它与每个迭代步矩阵 $\boldsymbol{F}'(\boldsymbol{x}^k)$的特征值有关，计算很困难，所以通常取 $\omega_k=\omega$，解法与解线性方程组 SOR 法相同. 当 $0<\omega<2$ 时，迭代收敛，迭代次数 j_k 可选为 $j_k=m, j_k=k+1$ 或 $j_k=2^k$ 等，也可根据 $\|\boldsymbol{x}^{k,j_k}-\boldsymbol{x}^{k,j_k-1}\|\leqslant\varepsilon$ 来控制. 若取 $j_k=1$，则式(9.5.6)可改写成

$$\boldsymbol{x}^{k+1}=\boldsymbol{x}^k-\omega_k(\boldsymbol{D}_k-\omega_k\boldsymbol{L}_k)^{-1}\boldsymbol{F}(\boldsymbol{x}^k)\quad(k=0,1,\cdots),\quad(9.5.7)$$

称为一步 N-SOR 迭代，它具有计算简单、计算量小的特点，但它只有线性敛速. 总的说，N-SOR 迭代(9.5.6)在当 $\omega_k\equiv\omega$ 且 $j_k=m$ 时，都只有线性敛速，只有当 j_k 依赖 k 序列$\{\boldsymbol{x}^k\}$时才可能具有超线性敛速.

9.5.2 非线性松弛迭代法

把解线性方程组的高斯-塞德尔(Gauss-Seidel)迭代法的思想直接用于解非线性方程组(9.1.1)，就可得到非线性松弛迭代法. 设 $\boldsymbol{x}^k=(x_1^k,\cdots,x_n^k)^{\mathrm{T}}$ 是方程(9.1.1)的 k 次近似解，则求 $\boldsymbol{x}^{k+1}$的第 i 个分量 x_i^{k+1}，可由 $\boldsymbol{F}=\boldsymbol{0}$ 的第 i 个分量方程

$$f_i(x_1^{k+1},\cdots,x_{i-1}^{k+1},x_i,x_{i+1}^k,\cdots,x_n^k)=0\quad(i=1,\cdots,n)\qquad(9.5.8)$$

的解 x_i 得到. 由 $\boldsymbol{x}^k$ 计算 $\boldsymbol{x}^{k+1}$的具体过程如下：

当 $i=1$ 时，由 $f_1(x_1,x_2^k,\cdots,x_n^k)=0$ 解出 x_1，记作 x_1^{k+1}. 当 $i=2$ 时，由$f_2(x_1^{k+1},x_2,x_3^k,\cdots,x_n^k)=0$ 解出 x_2，记作 x_2^{k+1}. 一般情况下，由方程(9.5.8)解出 x_i 记作 x_i^{k+1}. 由于每个分量方程为单变量非线性方程，所以，若用牛顿法求解，就得到高斯-塞德尔-牛顿迭代. 在求出方程(9.5.8)的解 x_i 后，引入松弛参数 $\omega_k>0$，

$$= \cdots$$

$$= H_k^{j_k} x^{k,0} + (H_k^{j_k - 1} + \cdots + H_k + I) B_k^{-1} b_k, \qquad (9.5.3)$$

注意到

$$B_k^{-1} A_k = B_k^{-1}(B_k - C_k) = I - H_k,$$

$$(H_k^{j_k-1} + \cdots + H_k + I)(I - H_k) = I - H_k^{j_k},$$

取 $x^{k,0} = x^k$，由式(9.5.3)可得

$$x^{k+1} = x^k - (I - H_k^{j_k}) x^k +$$

$$(H_k^{j_k-1} + \cdots + H_k + I) B_k^{-1} (A_k x^k - F(x^k))$$

$$= x^k - (H_k^{j_k-1} + \cdots + H_k + I) B_k^{-1} F(x^k) \quad (k = 0, 1, \cdots). \qquad (9.5.4)$$

这就是非线性——线性迭代法的一般格式. 若将 A_k 分裂为

$$A_k = D_k - L_k - U_k$$

其中 $D_k, -L_k, -U_k$ 分别为 A_k 的对角阵、严格下三角阵与严格上三角阵，并取 $B_k = D_k, C_k = L_k + U_k$. 当 D_k 非奇异时，则

$$H_k = D_k^{-1}(L_k + U_k).$$

代入式(9.5.4)，得牛顿-雅可比(Newton-Jacobi)迭代：

$$x^{k+1} = x^k - (H_k^{j_k-1} + \cdots + I) D_k^{-1} F(x^k) \quad (k = 0, 1, \cdots). \qquad (9.5.5)$$

如果引入松弛因子 $\omega_k \geqslant 0$，并取

$$B_k = \frac{1}{\omega_k}(D_k - \omega_k L_k), C_k = \frac{1}{\omega_k}[(1 - \omega_k) D_k + \omega_k U_k],$$

$$H_k = H_k(\omega_k) = B_k^{-1} C_k = (D_k - \omega_k L_k)^{-1}[(1 - \omega_k) D_k + \omega_k U_k],$$

代入式(9.5.4)，则得

$$x^{k+1} = x^k - \omega_k [H_k^{j_k-1}(\omega_k) + \cdots +$$

$$H_k(\omega_k) + I](D_k - \omega_k L_k)^{-1} F(x^k)$$

$$(k = 0, 1, \cdots). \qquad (9.5.6)$$

这种方法称为牛顿-SOR 迭代法. 当 $\omega_k = 1$ 时就是**牛顿-塞德尔**

并令

$$x_i^{k+1} = x_i^k + \omega_k(x_i - x_i^k) \quad (i = 1,\cdots,n) \qquad (9.5.9)$$

则称此迭代过程为**非线性 SOR 迭代**. 若求非线性方程(9.5.8)使用牛顿法,则称此迭代为 **SOR-牛顿法**. 它是双层迭代,主迭代为 SOR 法,内层迭代用牛顿法求 n 个单变量的非线性方程. 若用 m 步牛顿法,则得 m 步 SOR-牛顿迭代. 当 $m=1$ 时,可得一步 SOR-牛顿迭代

$$x_i^{k+1} = x_i^k - \omega_k \frac{f_i(x_1^{k+1},\cdots,x_{i-1}^{k+1},x_i^k,\cdots,x_n^k)}{\partial_i f_i(x_1^{k+1},\cdots,x_{i-1}^{k+1},x_i^k,\cdots,x_n^k)},$$

$$i = 1,2,\cdots,n; \quad k = 0,1,\cdots \qquad (9.5.10)$$

为了不计算偏导数,求单变量非线性方程解可用式(5.5.3)或式(5.5.2)的方法,从而得到 SOR-斯蒂芬森算法:

$$x_i^{k+1} = x_i^k - \omega_k \frac{(f_i^k)^2}{f_i(x_1^{k+1},\cdots,x_{i-1}^{k+1},x_i^k + f_i^k,x_{i+1}^k,\cdots,x_n^k) - f_i^k}$$

$$(i = 1,\cdots,n,k = 0,1,\cdots), \qquad (9.5.11)$$

其中

$$f_i^k = f_i(x_1^{k+1},\cdots,x_{i-1}^{k+1},x_i^k,\cdots,x_n^k) \quad (i = 1,\cdots,n) \qquad (9.5.12)$$

以及 SOR-割线法:

$$x_i^{k+1} = x_i^k - \omega_k \frac{(x_i^k - x_i^{k-1})f_i^k}{f_i^k - f_i(x_1^{k+1},\cdots,x_{i-1}^{k+1},x_i^{k-1},x_{i+1}^k,\cdots,x_n^k)}$$

$$(i = 1,2,\cdots,n;k = 1,2,\cdots), \qquad (9.5.13)$$

其中 f_i^k 仍由式(9.5.12)表示.

这些算法迭代一次只需计算 $2n$ 个分量函数值,相当于计算两个 $\boldsymbol{F}$ 值,并且不用解线性方程组,计算量小,适合于求解大型稀疏的非线性方程组,且便于做并行计算. 缺点是此方法只是线性收敛,效率较低.

9.6 割线法

不用计算导数的离散牛顿法实际上是一种割线法. 对单个方程,割线法优于牛顿法,但对方程组,割线法每步计算量与牛顿法相当,因此效率并不比牛顿法优. 单个方程的割线法是通过对函数 $y=f(x)$ 的线性插值导出的,即用曲线 $y=f(x)$ 上两点连线的割线近似曲线求割线与 x 轴交点得到. 对于 n 维情形,就是对空间 $\mathbb{R}^{n+1}$ 中的 n 个"分量超曲面"$y=f_i(\boldsymbol{x})(i=1,\cdots,n)$, 用 $\boldsymbol{x}^k$ 附近的 $n+1$ 个点 $\boldsymbol{x}^{k,j}(j=0,1,\cdots,n)$ 形成的 n 个"超割平面"近似代替,也就是用这 $n+1$ 个点作为插值节点构造一个仿射映射

$$\boldsymbol{L}_k(\boldsymbol{x}) = \boldsymbol{A}_k\boldsymbol{x} + \boldsymbol{b}_k, \tag{9.6.1}$$

其中 $\boldsymbol{L}_k(\boldsymbol{x})=(l_1(x),\cdots,l_n(x))^{\mathrm{T}}$, $\boldsymbol{A}_k\in\mathbb{R}^{n\times n}$, $\boldsymbol{b}_k\in\mathbb{R}^n$, 使

$$l_i(\boldsymbol{x}^{k,j}) = f_i(\boldsymbol{x}^{k,j}) \quad (i = 1,\cdots,n, j = 0,1,\cdots,n). \tag{9.6.2}$$

然后用线性方程组 $\boldsymbol{L}_k(\boldsymbol{x})=\boldsymbol{0}$ 的解 $\boldsymbol{x}^{k+1}$ 近似非线性方程组 $\boldsymbol{F}(\boldsymbol{x})=\boldsymbol{0}$ 的解 $\boldsymbol{x}^*$. 显然,必须选取这 $n+1$ 个点 $\boldsymbol{x}^{k,j}(j=0,1,\cdots,n)$ 以保证矩阵 $\boldsymbol{A}_k$ 非奇异. 从几何观点来说,就是要求这 $n+1$ 个点能张成空间 $\mathbb{R}^n$, 即 n 个向量 $\boldsymbol{x}^{k,j}-\boldsymbol{x}^{k,0}(j=1,\cdots,n)$ 必须线性无关,这时,称这 $n+1$ 个点处于**一般位置**.

定理 9.6.1 设给定 $\boldsymbol{x}^{k,j}\in\mathbb{R}^n(j=0,1,\cdots,n)$ 及 $\boldsymbol{F}: D\subset\mathbb{R}^n\to\mathbb{R}^n$, 则当且仅当 $\{\boldsymbol{x}^{k,j}\}$ 处于一般位置上时,存在惟一的仿射映射(9.6.1)使它满足条件(9.6.2). 此外,当且仅当 $\{\boldsymbol{F}(\boldsymbol{x}^{k,j})\}$ 处于一般位置上时,矩阵 $\boldsymbol{A}_k$ 才非奇异.

根据这个定理,可以构造求方程(9.1.1)的一般割线法. 设 $\boldsymbol{x}^{k,j}$ 及 $\boldsymbol{F}(\boldsymbol{x}^{k,j})(j=0,1,\cdots,n)$, 都处于一般位置上,若已知 $\boldsymbol{x}^k$ 为方程(9.1.1)的 k 次近似,取 $\boldsymbol{x}^{k,0}=\boldsymbol{x}^k$, 由条件(9.6.2)有

$$\boldsymbol{A}_k\boldsymbol{x}^k + \boldsymbol{b}_k = \boldsymbol{F}(\boldsymbol{x}^k),$$

$$\boldsymbol{A}_k\boldsymbol{x}^{k,j} + \boldsymbol{b}_k = \boldsymbol{F}(\boldsymbol{x}^{k,j}) \quad (j = 1,2,\cdots,n),$$

两式相减，得

$$\begin{cases} \boldsymbol{A}_k(\boldsymbol{x}^{k,j} - \boldsymbol{x}^k) = \boldsymbol{F}(\boldsymbol{x}^{k,j}) - \boldsymbol{F}(\boldsymbol{x}^k), \\ \boldsymbol{b}_k = \boldsymbol{F}(\boldsymbol{x}^k) - \boldsymbol{A}_k\boldsymbol{x}^k, \end{cases} \quad j = 1,\cdots,n \tag{9.6.3}$$

记为

$$\begin{cases} \boldsymbol{H}_k = (\boldsymbol{x}^{k,1} - \boldsymbol{x}^k, \boldsymbol{x}^{k,2} - \boldsymbol{x}^k, \cdots, \boldsymbol{x}^{k,n} - \boldsymbol{x}^k), \\ \boldsymbol{\Gamma}_k = (\boldsymbol{F}(\boldsymbol{x}^{k,1}) - \boldsymbol{F}(\boldsymbol{x}^k), \boldsymbol{F}(\boldsymbol{x}^{k,2}) - \boldsymbol{F}(\boldsymbol{x}^k), \cdots, \\ \qquad \boldsymbol{F}(\boldsymbol{x}^{k,n}) - \boldsymbol{F}(\boldsymbol{x}^k)). \end{cases} \tag{9.6.4}$$

由于$\{\boldsymbol{x}^{k,j}\}$在一般位置上，故 $\boldsymbol{H}_k$ 非奇异. 由式(9.6.3)可得

$$\boldsymbol{A}_k\boldsymbol{H}_k = \boldsymbol{\Gamma}_k, \quad \boldsymbol{A}_k = \boldsymbol{\Gamma}_k\boldsymbol{H}_k^{-1}. \tag{9.6.5}$$

由于$\{\boldsymbol{F}(\boldsymbol{x}^{k,j})\}$在一般位置上，故 $\boldsymbol{\Gamma}_k$ 非奇异，$\boldsymbol{A}_k$ 非奇异，且

$$\boldsymbol{A}_k^{-1} = \boldsymbol{H}_k\boldsymbol{\Gamma}_k^{-1},$$

于是方程(9.6.1)有解.

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \boldsymbol{A}_k^{-1}\boldsymbol{F}(\boldsymbol{x}^k) \text{ 或 } \boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \boldsymbol{H}_k\boldsymbol{\Gamma}_k^{-1}\boldsymbol{F}(\boldsymbol{x}^k),$$
$$k = 0,1,\cdots \tag{9.6.6}$$

这就是求非线性方程组的**一般割线法**. 割线法的计算量在很大程度上依赖于插值点的选取，选得适当可减少计算量并保证算法有较高收敛速度.

由式(9.6.4)中 $\boldsymbol{H}_k$ 的定义，可得

$$\boldsymbol{x}^{k,j} = \boldsymbol{x}^k + \boldsymbol{H}_k\boldsymbol{e}_j \quad (j = 1,\cdots,n), \tag{9.6.7}$$

其中 $\boldsymbol{e}_j$ 是第 j 个单位坐标向量. 若反过来考虑，任给一个非奇异矩阵 $\boldsymbol{H}_k \in \mathbb{R}^{n\times n}$，则由式(9.6.7)可确定 n 个点，这 n 个点连同 x^k 组成一组插值点. 显然，它们处于一般位置上，因此，给定的非奇异矩阵 $\boldsymbol{H}_k$ 等价于给定一组处于一般位置上的插值点，这样就可从构造 $\boldsymbol{H}$ 阵出发，形成各种割线法.

设给定 x 和非奇异矩阵 $\boldsymbol{H} \in \mathbb{R}^{n\times n}$，令

$$\boldsymbol{A}(\boldsymbol{x},\boldsymbol{H}) = \boldsymbol{\Gamma}\boldsymbol{H}^{-1},$$

其中

$$\boldsymbol{\Gamma}=(\boldsymbol{F}(\boldsymbol{x}+\boldsymbol{H}\boldsymbol{e}_1)-\boldsymbol{F}(\boldsymbol{x}),\cdots,\boldsymbol{F}(\boldsymbol{x}+\boldsymbol{H}\boldsymbol{e}_n)-\boldsymbol{F}(\boldsymbol{x})),$$

则得割线法一般形式为

$$\boldsymbol{x}^{k+1}=\boldsymbol{x}^k-\boldsymbol{A}(\boldsymbol{x}^k,\boldsymbol{H}_k)^{-1}\boldsymbol{F}(\boldsymbol{x}^k)\quad(k=0,1,\cdots).\tag{9.6.8}$$

如取

$$\boldsymbol{H}_k=\mathrm{diag}(h_1^k,h_2^k,\cdots,h_n^k)\quad(h_j^k\neq 0,j=1,\cdots,n),$$

则

$$\boldsymbol{\Gamma}_k=[\boldsymbol{F}(\boldsymbol{x}^k+h_1^k\boldsymbol{e}_1)-\boldsymbol{F}(\boldsymbol{x}^k),\cdots,\boldsymbol{F}(\boldsymbol{x}^k+h_n^k\boldsymbol{e}_n)-\boldsymbol{F}(\boldsymbol{x}^k)],$$

$$\boldsymbol{A}(\boldsymbol{x}^k,\boldsymbol{H}_k)=\boldsymbol{\Gamma}_k\boldsymbol{H}_k^{-1}=\Big(\frac{1}{h_1^k}[\boldsymbol{F}(\boldsymbol{x}^k+h_1^k\boldsymbol{e}_1)-\boldsymbol{F}(\boldsymbol{x}^k)],\cdots,$$

$$\frac{1}{h_n^k}[\boldsymbol{F}(\boldsymbol{x}^k+h_n^k\boldsymbol{e}_n)-\boldsymbol{F}(\boldsymbol{x}^k)]\Big).\tag{9.6.9}$$

上式确定的 $\boldsymbol{A}(\boldsymbol{x}^k,\boldsymbol{H}_k)$ 正是 $\boldsymbol{F}(\boldsymbol{x})$ 的雅可比阵 $\boldsymbol{F}'(\boldsymbol{x})$ 的离散形式，由此得到的割线法就是离散牛顿法(9.3.13). 当 $h_j^k=f_j(\boldsymbol{x}^k)$ 时，即是式(9.3.14)的牛顿-斯蒂芬森程序.

若取 $h_j^k=x_j^{k-1}-x_j^k(j=1,2,\cdots,n)$，即插值点只与 $\boldsymbol{x}^{k-1}$ 及 $\boldsymbol{x}^k$ 有关，则此时

$$\boldsymbol{A}(\boldsymbol{x}^k,\boldsymbol{H}_k)=\Big(\frac{1}{x_1^{k-1}-x_1^k}[\boldsymbol{F}(\boldsymbol{x}^k+(x_1^{k-1}-x_1^k)\boldsymbol{e}_1)-\boldsymbol{F}(\boldsymbol{x}^k)],\cdots,$$

$$\frac{1}{x_n^{k-1}-x_n^k}[\boldsymbol{F}(\boldsymbol{x}^k+(x_n^{k-1}-x_n^k)\boldsymbol{e}_n)-\boldsymbol{F}(\boldsymbol{x}^k)]\Big).\tag{9.6.10}$$

对应的割线法(9.6.8)就称为**两点割线法**. 它只要给定两上初始近似 $\boldsymbol{x}^0$ 及 $\boldsymbol{x}^1$，即可按式(9.6.8)及式(9.6.10)的格式进行迭代. 若取

$$\boldsymbol{H}_k=\Big(h_1^k\boldsymbol{e}_1,\sum_{i=1}^{2}h_i^k\boldsymbol{e}_i,\cdots,\sum_{i=1}^{n}h_i^k\boldsymbol{e}_i\Big)=\begin{pmatrix}h_1^k & h_1^k & \cdots & h_1^k\\ & h_2^k & \cdots & h_2^k\\ & & \ddots & \vdots\\ & & & h_n^k\end{pmatrix},$$

其中 $h_j^k = x_j^{k-1} - x_j^k$，则

$$\boldsymbol{\Gamma}_k = \left[\boldsymbol{F}(\boldsymbol{x}^k + h_1^k\boldsymbol{e}_1) - \boldsymbol{F}(\boldsymbol{x}^k), \cdots, \boldsymbol{F}\left(\boldsymbol{x}^k + \sum_{i=1}^n h_i^k\boldsymbol{e}_i\right) - \boldsymbol{F}(\boldsymbol{x}^k)\right].$$

若引进矩阵

$$\boldsymbol{P} = \begin{pmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \\ & & & & 1 \end{pmatrix},$$

则

$$\boldsymbol{H}_k\boldsymbol{P} = \mathrm{diag}(h_1^k, \cdots, h_n^k),$$

$$\boldsymbol{\Gamma}_k\boldsymbol{P} = \left[\boldsymbol{F}(\boldsymbol{x}^k + h_1^k\boldsymbol{e}_1) - \boldsymbol{F}(\boldsymbol{x}^k), \boldsymbol{F}\left(\boldsymbol{x}^k + \sum_{i=1}^2 h_i^k\boldsymbol{e}_k\right) - \right.$$

$$\left.\boldsymbol{F}(\boldsymbol{x}^k + h_1^k\boldsymbol{e}_1), \cdots, \boldsymbol{F}\left(\boldsymbol{x}^k + \sum_{i=1}^n h_i^k\boldsymbol{e}_i\right) - \boldsymbol{F}\left(\boldsymbol{x}^k + \sum_{i=1}^{n-1} h_i^k\boldsymbol{e}_i\right)\right].$$

于是

$$\begin{aligned}\boldsymbol{A}(\boldsymbol{x}^k, \boldsymbol{H}^k) &= \boldsymbol{\Gamma}_k\boldsymbol{H}_k^{-1} = (\boldsymbol{\Gamma}_k\boldsymbol{P})(\boldsymbol{H}_k\boldsymbol{P})^{-1} \\ &= \left[\frac{1}{h_1^k}(\boldsymbol{F}(\boldsymbol{x}^k + h_1^k\boldsymbol{e}_1) - \boldsymbol{F}(\boldsymbol{x}^k)), \cdots, \frac{1}{h_n}\left(\boldsymbol{F}\left(\boldsymbol{x}^k + \right.\right.\right. \\ &\left.\left.\left.\sum_{i=1}^n h_i^k\boldsymbol{e}_i\right) - \boldsymbol{F}\left(\boldsymbol{x}^k + \sum_{i=1}^{n-1} h_i^k\boldsymbol{e}_i\right)\right)\right]. \qquad (9.6.11)\end{aligned}$$

这样得到的割线法(9.6.8)与式(9.6.11)是另一种形式的两点割线法，它每步也要算 n 个函数值(即 n^2 个分量函数值). 对一点割线法即离散牛顿法，适当选取 $h_j^k(j=1,\cdots,n)$，生成的序列 $\{\boldsymbol{x}^k\}$ 可达到平方敛速. 对两点割线法则有以下收敛定理.

定理 9.6.2 假定 $\boldsymbol{F}: D\subset\mathbb{R}^n \to \mathbb{R}^n$ 在 $\boldsymbol{x}^*$ 的开邻域 $S_0 = S(\boldsymbol{x}^*, \delta_0)\subset D$ 连续可导，$\boldsymbol{F}(\boldsymbol{x}^*) = \boldsymbol{0}$，且 $\boldsymbol{F}'(\boldsymbol{x}^*)$ 非奇异，则存在球 $S = S(\boldsymbol{x}^*, \delta)\subset S_0$，使对任何 $\boldsymbol{x}^0, \boldsymbol{x}^1 \in S$，由两点割线法生成的迭代序列

$\{x^k\}\subset S$，且超线性收敛于 x^*. 如果存在 $\gamma>0$，使

$$\| F'(x) - F'(y) \| \leqslant \gamma \| x - y \|, \quad \forall x, y \in S_0$$

成立，则此序列 $\{x^k\}$ 的收敛阶 $p_R \geqslant \tau_1 = \frac{1+\sqrt{5}}{2} \approx 1.618$，这里 τ_1 是方程 $t^2-t-1=0$ 的惟一正根.

由此可知，两点割线法的效率

$$e(S_1) = \frac{1}{n}\ln\left(\frac{1+\sqrt{5}}{2}\right).$$

当 $n\geqslant 3$ 时，

$$e(S_1) < e(N_1) = \frac{\ln 2}{n+1},$$

这里 $e(N_1)$ 是牛顿法效率. 为提高两点割线法效率，可对它使用沙曼斯基技巧，这就是布兰特方法的另一种算法. 只要由 x^{k-1} 及 x^k 确定一个非奇阵 H_k，就可得到两点割线法. 现要求 H_k 为正交阵，取 $H_k = h_k Q_k$，其中 Q_k 是由向量 $x^{k-1}-x^k$ 找到的一个镜像变换阵，使

$$x^{k-1} - x^k = h_k Q_k e_1,$$

其中 $h_k = \| x^{k-1} - x^k \|_2$. 这是容易做到的，事实上，对任意向量 $x\in\mathbb{R}^n$. 记 $\sigma = \pm \| x \|_2$，设 $x \neq -\sigma e_1$，则必存在一个镜像变换阵 U，使 $Ux = -\sigma e_1$. 可以证明

$$U = I - 2\frac{uu^{\mathrm{T}}}{\| u \|_2^2},$$

其中 $u = x + \sigma e_1$.

根据这一性质，由 $x^{k-1}-x^k$ 可求得

$$Q_k = I - 2\frac{u_k u_k^{\mathrm{T}}}{\| u_k \|_2^2},$$

其中 $u_k = x^{k-1} - x^k \pm h_k e_1$，当 $x_1^{k-1} - x_1^k \geqslant 0$ 时取"+"号. 否则取"−"号，于是可得下列算法.

算法 9.6.3 给定 F: $D\subset\mathbb{R}^n\to\mathbb{R}^n$ 及两个不同初始近似 x^0，

$\overline{x^0}\in D$,以及 $k_0\geqslant 1$,然后按下列步骤求解:

(1) 确定正交矩阵 Q_k,使

$$\bar{x}^k = x^k + h_k Q_k e_1 \quad (h_k = \| x^k - x^{k-1} \|_2).$$

(2) 计算矩阵

$$\Gamma_k = \frac{1}{h_k}[F(x^k + h_k Q_k e_1) - F(x^k), \cdots, F(x^k + h_k Q_k e_k) - F(x^k)].$$

(注意:这里 $F(\bar{x}^k) = F(x^k + h_k Q_k e_1)$ 已知,故可节省计算 F 值一次.)

(3) 以 $A_k = \Gamma_k Q_k$ 进行 k_0 次迭代计算,即令 $y^{k,0} = x^k$ 并计算

$$y^{k,l} = y^{k,l-1} - A_k^{-1} F(y^{k,l-1}) \quad (l = 1, \cdots, k_0).$$

(4) 置 $x^{k+1} = y^{k,k_0}$,$\bar{x}^{k+1} = y^{k,k_0-1}$,若 x^{k+1} 满足精度要求,则置 $x^* = x^{k+1}$,并停止迭代,否则执行步骤(5).

(5) 置 $k = k+1$,转步骤(1).

以上算法称为布兰特的 S_{k_0} 算法. 当 $k_0 = 1$ 即为两点割线法(9.6.8)与(9.6.10). S_{k_0} 算法每迭代一步需计算 $n + k_0 - 1$ 次函数值.

在定理 9.6.2 的条件下,S_{k_0} 算法生成的序列$\{x^k\}$收敛于 x^*,其收敛阶至少为

$$p(k_0) = \frac{k_0 + \sqrt{k_0^2 + 4}}{2}.$$

由以上结果可知,S_{k_0} 算法的效率为

$$e(S_{k_0}) = \frac{\ln p(k_0)}{n + k_0 - 1}.$$

对固定的 n,可求最优的$\overline{k_0}$,使

$$e(S_{\bar{k}_0}) = \max_{k_0} \frac{\ln p(k_0)}{n + k_0 - 1}.$$

下表列出某些固定 n 时,$\bar{k}_0$ 及效率比$\dfrac{e(S_{\bar{k}_0})}{e(N_1)}$的值.

表 9.4

n	2	3	5	10	20	50	100	1000
$\overline{k_0}$	3	4	5	8	12	23	38	226
$e(S_{k_0})/e(N_1)$	1.29	1.39	1.58	1.96	2.44	3.21	3.87	6.39

9.7 拟牛顿法

9.7.1 拟牛顿法的基本思想

牛顿法、离散牛顿法及割线法都可表示为

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \boldsymbol{A}_k^{-1}\boldsymbol{F}(\boldsymbol{x}^k) \quad (k = 0,1,\cdots), \qquad (9.7.1)$$

它们的计算量较大,这使得如何减少计算量并保持较高的收敛速度成为求解非线性方程组一个十分重要的问题.拟牛顿法就是针对这一问题提出的新方法,它用 $\boldsymbol{A}_k$ 近似 $\boldsymbol{F}'(\boldsymbol{x}^k)$,而 $\boldsymbol{A}_{k+1}$ 可在 $\boldsymbol{A}_k$ 的基础上用一个低秩的矩阵 $\Delta\boldsymbol{A}_k$ 来校正,这样既减少了计算函数值次数,又使求 $\boldsymbol{A}_k^{-1}$ 运算量降为 $O(n^2)$次算术运算.

设 $\boldsymbol{F}: D\subset\mathbb{R}^n\to\mathbb{R}^n$连续可导,$\boldsymbol{s}_k=\boldsymbol{x}^{k+1}-\boldsymbol{x}^k\neq\boldsymbol{0},\boldsymbol{x}^k,\boldsymbol{x}^{k+1}\in D$,给定 $\epsilon>0$,$\exists\,\delta>0$,当 $\|\boldsymbol{s}_k\|<\delta$ 时,

$$\|\boldsymbol{F}(\boldsymbol{x}^k) - \boldsymbol{F}(\boldsymbol{x}^{k+1}) - \boldsymbol{F}'(\boldsymbol{x}^{k+1})(\boldsymbol{x}^k - \boldsymbol{x}^{k+1})\| \leqslant \epsilon\|\boldsymbol{s}_k\|$$

成立.因此有

$$\boldsymbol{F}(\boldsymbol{x}^k) \approx \boldsymbol{F}(\boldsymbol{x}^{k+1}) + \boldsymbol{F}'(\boldsymbol{x}^{k+1})(\boldsymbol{x}^k - \boldsymbol{x}^{k+1}),$$

其近似程度随 $\|\boldsymbol{x}^k - \boldsymbol{x}^{k+1}\|$ 减少而增大.若以 $\boldsymbol{A}_{k+1}$ 近似代替 $\boldsymbol{F}'(\boldsymbol{x}^{k+1})$,则要求 $\boldsymbol{A}_{k+1}$满足方程

$$\boldsymbol{A}_{k+1}(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k) = \boldsymbol{F}(\boldsymbol{x}^{k+1}) - \boldsymbol{F}(\boldsymbol{x}^k). \qquad (9.7.2)$$

当 $n=1$ 时,方程(9.7.2)为

$$\boldsymbol{A}_{k+1} = \frac{\boldsymbol{F}(\boldsymbol{x}^{k+1}) - \boldsymbol{F}(\boldsymbol{x}^k)}{\boldsymbol{x}^{k+1} - \boldsymbol{x}^k},$$

它是 $\boldsymbol{F}(\boldsymbol{x})$关于点 $\boldsymbol{x}^{k+1}$与 $\boldsymbol{x}^k$ 的差商.当 $n>1$ 时表明矩阵 $\boldsymbol{A}_{k+1}$关于

点 $\boldsymbol{x}^k$ 与 $\boldsymbol{x}^{k+1}$ 具有“差商”性质，称式(9.7.2)为广义差商条件. 因为 $\boldsymbol{A}_{k+1}$ 可以看作 $\boldsymbol{F}'(\boldsymbol{x}^{k+1})$ 的近似矩阵，故式(9.7.2)又称为拟牛顿方程. 由于 $\boldsymbol{A}_{k+1}$ 有 n^2 个元素，而方程(9.7.2)只给出 n 个条件，所以当 $n>1$ 时，A_{k+1} 不能完全确定，它还有很大的选择余地. 如果 $\boldsymbol{A}_{k+1}$ 能表示为 $\boldsymbol{A}_k+\Delta\boldsymbol{A}_k$（$\Delta\boldsymbol{A}_k$ 为低秩矩阵），则可得到一类迭代算法：

$$\begin{cases}\boldsymbol{x}^{k+1}=\boldsymbol{x}^k-\boldsymbol{A}_k^{-1}\boldsymbol{F}(\boldsymbol{x}^k),\\ \boldsymbol{A}_{k+1}(\boldsymbol{x}^{k+1}-\boldsymbol{x}^k)=\boldsymbol{F}(\boldsymbol{x}^{k+1})-\boldsymbol{F}(\boldsymbol{x}^k), \quad k=0,1,\cdots \\ \boldsymbol{A}_{k+1}=\boldsymbol{A}_k+\Delta\boldsymbol{A}_k, \mathrm{rank}\Delta\boldsymbol{A}_k=m\geqslant 1,\end{cases} \tag{9.7.3}$$

称为**拟牛顿法**，或称为**秩 m 校正方法**，如果对所有 k，$\boldsymbol{A}_k^{-1}=\boldsymbol{H}_k$ 存在，则迭代(9.7.3)是完全确定的. 利用 Sherman-Morrison-Woodbury 公式(9.7.4)，可得到 $\Delta\boldsymbol{H}_k$ 的一个显式表达式.

Sherman-Morrison-Woodbury 公式如下：设 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 非奇异，$\boldsymbol{U},\boldsymbol{V}\in\mathbb{R}^{n\times m}$，$m\leqslant n$，则当且仅当 $\boldsymbol{I}+\boldsymbol{V}^{\mathrm{T}}\boldsymbol{A}^{-1}\boldsymbol{U}$ 非奇异时，$\boldsymbol{A}+\boldsymbol{U}\boldsymbol{V}^{\mathrm{T}}$ 也非奇异，且

$$(\boldsymbol{A}+\boldsymbol{U}\boldsymbol{V}^{\mathrm{T}})^{-1}=\boldsymbol{A}^{-1}-\boldsymbol{A}^{-1}\boldsymbol{U}(\boldsymbol{I}+\boldsymbol{V}^{\mathrm{T}}\boldsymbol{A}^{-1}\boldsymbol{U})^{-1}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{A}^{-1}. \tag{9.7.4}$$

因为 $\mathrm{rank}(\Delta\boldsymbol{A}_k)=m$，而任何秩 m 的 n 阶矩阵都可分解为 $\Delta\boldsymbol{A}_k=\boldsymbol{U}_k\boldsymbol{V}_k^{\mathrm{T}}$，$\boldsymbol{U}_k,\boldsymbol{V}_k\in\mathbb{R}^{n\times m}$，$\mathrm{rank}\boldsymbol{U}_k=\mathrm{rank}\boldsymbol{V}_k=m$，所以，如果$(\boldsymbol{I}+\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{A}_k^{-1}\boldsymbol{U}_k)$非奇异，则

$$\begin{aligned}\boldsymbol{H}_{k+1}&=(\boldsymbol{A}_k+\Delta\boldsymbol{A}_k)^{-1}\\&=\boldsymbol{A}_k^{-1}-\boldsymbol{A}_k^{-1}\boldsymbol{U}_k(\boldsymbol{I}+\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{A}_k^{-1}\boldsymbol{U}_k)^{-1}\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{A}_k^{-1}\\&=\boldsymbol{H}_k+\Delta\boldsymbol{H}_k,\end{aligned}$$

其中

$$\Delta\boldsymbol{H}_k=\boldsymbol{Y}_k\boldsymbol{W}_k^{\mathrm{T}},$$

$$\boldsymbol{Y}_k=-\boldsymbol{A}_k^{-1}\boldsymbol{U}_k(\boldsymbol{I}+\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{A}_k^{-1}\boldsymbol{U}_k)^{-1}, \quad \boldsymbol{W}_k=(\boldsymbol{A}_k^{-1})^{\mathrm{T}}\boldsymbol{V}_k.$$

显然，$\boldsymbol{Y}_k,\boldsymbol{W}_k\in\mathbb{R}^{n\times m}$，且它们的秩为 m，故 $\Delta\boldsymbol{H}_k$ 仍为一个秩 m 矩

阵. 这样, 式(9.7.3)又可写成

$$\begin{cases} x^{k+1} = x^k - H_k F(x^k), \\ H_{k+1}[F(x^{k+1}) - F(x^k)] = x^{k+1} - x^k, \quad k = 0,1,\cdots \\ H_{k+1} = H_k + \Delta H_k, \quad \text{rank}\Delta H_k = m, \end{cases} \tag{9.7.5}$$

这就是逆秩 m 校正公式.

不同的计算方案对应不同算法, 常用的算法只是 $m=1$ 及 $m=2$ 的公式, 即使用秩 1 及秩 2 校正公式.

9.7.2 布罗依登方法

布罗依登(Broyden)于 1965 年首先提出了一种秩 1 的校正公式: 由 x^k 计算 x^{k+1} 只计算一次 F 函数值及做 $O(n^2)$ 次算术运算. 为书写方便, 把上下标去掉, 用 $\bar{x}$ 表示由 x 出发得到的新近似, 相应地, 由 A 得到 $\bar{A}$, H 得到 $\bar{H}$, 迭代过程(9.7.1)可写成

$$\bar{x} = x - A^{-1}F(x).$$

拟牛顿方程(9.7.2)可写成

$$\bar{A}s = y, \tag{9.7.6}$$

其中 $s=\bar{x}-x$, $y=F(\bar{x})-F(x)$. 布罗依登方法要求 $\bar{A}$ 与 A 在向量 s 的正交补 $s^{\perp}$ 上两者无任何差别, 即对 $\forall z\in s^{\perp}$, $\bar{A}z=Az$, 也就是

$$(\bar{A} - A)z = 0$$

对 $\forall (z,s)=s^{\mathrm{T}}z=0$ 成立, 故 $\bar{A}-A$ 为秩 1 的矩阵. 可设

$$\bar{A} - A = us^{\mathrm{T}},$$

这里 $u\in\mathbb{R}^n$ 为待定向量. 代入上式则得

$$us^{\mathrm{T}}z = 0.$$

由方程(9.7.6)可得

$$(\bar{A} - A)s = y - As,$$

于是

$$us^{\mathrm{T}}s = y - As,$$

$$u = \frac{y - As}{s^T s},$$

从而推得

$$\bar{A} = A + \frac{(y - As)s^T}{s^T s} \quad (s \neq 0). \tag{9.7.7}$$

当 $s=0$ 时，$\bar{x}=x$，迭代即终止，式(9.7.7)就是秩 1 的校正公式，这个公式具有一个重要性质，即在所有满足拟牛顿方程的矩阵中，$\bar{A}$ 是在弗罗比尼乌斯(Frobenius)范数意义下最接近 A 的矩阵.

定理 9.7.1 设给定 $A \in \mathbb{R}^{n\times n}$，$y \in \mathbb{R}^n$ 和非零向量 $s \in \mathbb{R}^n$，则由式(9.7.7)确定的矩阵 $\bar{A}$ 是以下问题的惟一解：

$$\min\{\, \|\hat{A} - A\|_F \mid \hat{A}s = y \}.$$

由公式(9.7.7)与拟牛顿法(9.7.1)得

$$\begin{cases} x^{k+1} = x^k - A_k^{-1} F(x^k), \\ A_{k+1} = A_k + \dfrac{(y^k - A_k s_k)s_k^T}{s_k^T s_k}, \end{cases} \quad k = 0,1,\cdots \tag{9.7.8}$$

其中 $s_k = x^{k+1} - x^k$，$y^k = F(x^{k+1}) - F(x^k)$. 公式(9.7.8)称为 Broyden 方法.

满足拟牛顿方程(9.7.6)的秩 1 算法有很多不同选法，如果要求 $\bar{A}$ 与 A 在向量 c 的正交补 $c^\perp$ 上两者无任何差别，则可得 $\Delta A = \bar{A} - A = uc^T$. 由

$$(\bar{A} - A)s = y - As = F(\bar{x}),$$

可得 $uc^T s = F(\bar{x})$，即 $u = \dfrac{F(\bar{x})}{c^T s}$，于是

$$\bar{A} = A + \frac{F(\bar{x})c^T}{c^T s}. \tag{9.7.9}$$

在公式(9.7.9)中取不同的 c，即可得到不同的秩 1 校正公式. 若 $c=s$，则可得式(9.7.1)；若取 $c = F(\bar{x}) = F(x^{k+1})$，则可得另一秩 1 拟牛顿法：

$$\begin{cases} x^{k+1} = x^k - A_k^{-1}F(x^k), \\ A_{k+1} = A_k + \dfrac{F(x^{k+1})F(x^{k+1})^{\mathrm{T}}}{F(x^{k+1})^{\mathrm{T}}(x^{k+1} - x^k)}, \end{cases} \tag{9.7.10}$$

这种方法称为**布罗依登第二方法**. 其特点是 ΔA_k 为对称矩阵,因此,它适合于对 F 的雅可比矩阵为对称的方程组求解,这时初始矩阵 A_0 也应取为对称矩阵.

由此得到的**布罗依登**方法(9.7.8)及(9.7.10),计算都较简单,对给定的 x^k 及 A_k,每步只需算一个新的 F 函数值,然后求解 $A_ks_k = -F(x^k)$ 得到 x^{k+1},再计算 $F(x^{k+1})$,从而求得 A_{k+1},表面上看,解方程要用 $O(n^3)$ 的算术运算,但如通过 QR 分解,可使运算量降为 $O(n^2)$. 通常可设 $A_k = Q_kR_k$,其中 Q_k 为正交阵,R_k 为上三角阵,由此形成 A_{k+1} 的 QR 分解,只需 $O(n^2)$ 算术运算.

对校正公式(9.7.7),利用 Sherman-Morrison 公式,有

$$(A + uv^{\mathrm{T}})^{-1} = A^{-1} - \frac{1}{\sigma}(A^{-1}uv^{\mathrm{T}}A^{-1}),$$

其中 $A \in \mathbb{R}^{n\times n}$ 非奇异,$u, v \in \mathbb{R}^n$,$\sigma = 1 + v^{\mathrm{T}}A^{-1}u \neq 0$.

令 $H = A^{-1}$,$\bar{H} = \bar{A}^{-1}$,$v = s$,$u = \dfrac{y - As}{s^{\mathrm{T}}s}$,则得

$$\bar{H} = H - \frac{Hus^{\mathrm{T}}H}{1 + s^{\mathrm{T}}Hu} = H + \frac{(s - Hy)s^{\mathrm{T}}H}{s^{\mathrm{T}}Hy}, \tag{9.7.11}$$

其中 $s^{\mathrm{T}}Hy \neq 0$,这是 $\bar{H}$ 非奇异的充要条件. 由式(9.7.11)可得到**逆布罗依登秩 1 公式**

$$\begin{aligned} x^{k+1} &= x^k - H_kF(x^k), \\ H_{k+1} &= H_k + \frac{(s_k - H_ky^k)s_k^{\mathrm{T}}H_k}{s_k^{\mathrm{T}}H_ky^k} \quad (k = 0, 1, \cdots). \end{aligned} \tag{9.7.12}$$

对应于式(9.7.10)的**逆布罗依登秩 1 第二公式**为

$$\begin{cases} x^{k+1} = x^k - H_kF(x^k), \\ H_{k+1} = H_k + (s_k - H_ky^k)\dfrac{(s_k - H_ky^k)^{\mathrm{T}}}{(s_k - H_ky^k)^{\mathrm{T}}y^k}, \end{cases} \quad k = 0, 1, \cdots \tag{9.7.13}$$

在一定条件下，可以证明布罗依登秩1方法生成的序列$\{x^k\}$是局部超线性收敛的.

例 9.7.2 用布罗依登方法(9.7.12)求解方程组

$$\begin{cases} f_1(x_1,x_2) = x_1^2 - x_2 - 1 = 0, \\ f_2(x_1,x_2) = (x_1-2)^2 + (x_2-0.5)^2 - 1 = 0. \end{cases}$$

初始近似取为$x^0=(0,0)^T$.

解 先计算$F(x^0)=(-1,3.25)^T$.因为

$$F'(x) = \begin{bmatrix} 2x_1 & -1 \\ 2x_1-4 & 2x_2-1 \end{bmatrix},$$

故

$$F'(x^0) = \begin{pmatrix} 0 & -1 \\ -4 & -1 \end{pmatrix},$$

$$H_0 = F'(x^0)^{-1} = \begin{pmatrix} 0.25 & -0.25 \\ -1 & 0 \end{pmatrix}.$$

由公式(9.7.12)可算得

$$s_0 = (1.0625, -1)^T, \quad x^1 = (1.0625, -1)^T,$$

$$F(x^1) = (1.12890625, 2.12890625)^T,$$

$$y^1 = F(x^1) - F(x^0) = \begin{pmatrix} 2.12890625 \\ -1.12109375 \end{pmatrix},$$

$$H_1 = \begin{pmatrix} 0.3557441 & -0.2721932 \\ -0.5224991 & -0.1002162 \end{pmatrix},$$

由此再算得

$$x^2 = (1.240372062, -0.196797467)^T.$$

按上述算法，经11次迭代可得到具有12位有效数字的近似解

$$x^{11} = (1.54634288332, 1.39117631279)^T.$$

如用牛顿法，则只要7次迭代即可达到同样精度的近似解.布罗依登方法虽然比牛顿法迭代多4次，但它每步计算量比牛顿法少得

多，所以，单从理论上还不能断定这两种方法效率的高低.

9.7.3 秩 2 校正公式

在式(9.7.3)与式(9.7.5)中，若取 $m=2$，则可得到一系列公式，这就是秩 2 校正公式. 在计算中，通常以逆拟牛顿公式(9.7.5)使用较多，为了具体得到 $\Delta\boldsymbol{H}_k$ 表达式，设 $\boldsymbol{Y}_k,\boldsymbol{W}_k\in\mathbb{R}^{n\times 2}$，并分别表示为

$$\boldsymbol{Y}_k=[\boldsymbol{y}_k^1,\boldsymbol{y}_k^2] \text{ 及 } \boldsymbol{W}_k=[\boldsymbol{w}_k^1,\boldsymbol{w}_k^2],$$

其中 $\boldsymbol{y}_k^i,\boldsymbol{w}_k^i\in\mathbb{R}^n(i=1,2)$. 将它代人逆拟牛顿公式

$$\boldsymbol{H}_{k+1}\boldsymbol{y}^k=\boldsymbol{s}_k, \tag{9.7.14}$$

由于 $\Delta\boldsymbol{H}_k=\boldsymbol{Y}_k\boldsymbol{W}_k^{\mathrm{T}}$，则式(9.7.14)可写成

$$(\boldsymbol{H}_k+\boldsymbol{y}_k^1(\boldsymbol{w}_k^1)^{\mathrm{T}}+\boldsymbol{y}_k^2(\boldsymbol{w}_k^2)^{\mathrm{T}})\boldsymbol{y}^k=\boldsymbol{s}_k. \tag{9.7.15}$$

若记 $\boldsymbol{r}_i=(\boldsymbol{w}_k^i)^{\mathrm{T}}\boldsymbol{y}^k(i=1,2)$，则式(9.7.15)可写成

$$\boldsymbol{r}_1\boldsymbol{y}_k^1+\boldsymbol{r}_2\boldsymbol{y}_k^2=\boldsymbol{s}_k-\boldsymbol{H}_k\boldsymbol{y}^k.$$

当 $\boldsymbol{r}_i\neq\boldsymbol{0}(i=1,2)$ 时，可取

$$\boldsymbol{y}_k^1=\left(\frac{1}{r_1}\right)\boldsymbol{s}_k,\quad \boldsymbol{y}_k^2=-\left(\frac{1}{r_2}\right)\boldsymbol{H}_k\boldsymbol{y}^k,$$

显然这时式(9.7.14)满足，于是得

$$\boldsymbol{H}_{k+1}=\boldsymbol{H}_k+\frac{\boldsymbol{s}_k(\boldsymbol{w}_k^1)^{\mathrm{T}}}{(\boldsymbol{w}_k^1)^{\mathrm{T}}\boldsymbol{y}^k}-\frac{\boldsymbol{H}_k\boldsymbol{y}^k(\boldsymbol{w}_k^2)^{\mathrm{T}}}{(\boldsymbol{w}_k^2)^{\mathrm{T}}\boldsymbol{y}^k}\quad(k=0,1,\cdots). \tag{9.7.16}$$

通过计算表明，当且仅当

$$(\boldsymbol{w}_k^2)^{\mathrm{T}}\boldsymbol{F}(\boldsymbol{x}^k)\neq\boldsymbol{0}\quad(k=0,1,\cdots)$$

时，非奇异条件 $(\boldsymbol{I}+\boldsymbol{w}_k^{\mathrm{T}}\boldsymbol{\Lambda}_k\boldsymbol{y}_k)^{-1}$ 成立. 因此，只要 $\boldsymbol{w}_k^i$ 满足

$$(\boldsymbol{w}_k^i)^{\mathrm{T}}\boldsymbol{y}^k\neq\boldsymbol{0}\quad(i=1,2),$$
$$(\boldsymbol{w}_k^2)^{\mathrm{T}}F(x^k)\neq 0\quad(k=0,1,\cdots),$$

则算法(9.7.16)生成一个非奇异矩阵序列 $\{\boldsymbol{H}_k\}$，它满足方程 $(\boldsymbol{H}_{k+1}-\boldsymbol{H}_k)\boldsymbol{y}^k=-\boldsymbol{H}_k\boldsymbol{F}(\boldsymbol{x}^{k+1})$，因此，只要选择不同的 $\boldsymbol{w}_k^1$ 及 $\boldsymbol{w}_k^2$ 就可产生不同的校正公式. 当 $\boldsymbol{w}_k^1$ 及 $\boldsymbol{w}_k^2$ 线性相关时，则得秩 1 校正公

式；当 w_k^1 及 w_k^2 线性无关时，则可得秩 2 校正公式. 通常，较好的秩 2 算法有如下几种：

1. DFP(Davidon-Fletcher-Powell)秩 2 算法. 在式(9.7.16)中取

$$\boldsymbol{w}_k^1 = \boldsymbol{s}_k, \boldsymbol{w}_k^2 = \boldsymbol{H}_k^{\mathrm{T}} \boldsymbol{y}^k \quad (k = 0,1,\cdots),$$

可得 DFP 算法计算公式

$$\begin{cases} \boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \boldsymbol{H}_k \boldsymbol{F}(\boldsymbol{x}^k), \\ \boldsymbol{H}_{k+1} = \boldsymbol{H}_k + \dfrac{\boldsymbol{s}_k \boldsymbol{s}_k^{\mathrm{T}}}{(\boldsymbol{s}_k)^{\mathrm{T}} \boldsymbol{y}^k} - \dfrac{\boldsymbol{H}_k \boldsymbol{y}^k (\boldsymbol{y}^k)^{\mathrm{T}} \boldsymbol{H}_k}{(\boldsymbol{y}^k)^{\mathrm{T}} \boldsymbol{H}_k \boldsymbol{y}^k}, \end{cases} \quad k = 0,1,\cdots. \tag{9.7.17}$$

2. BFS(Broyden-Fletcher-Shanno)秩 2 算法. 在式(9.7.16)中，取

$$\frac{(\boldsymbol{w}_k^1)^{\mathrm{T}}}{(\boldsymbol{w}_k^1)^{\mathrm{T}} \boldsymbol{y}^k} = \mu_k \frac{\boldsymbol{s}_k^{\mathrm{T}}}{\boldsymbol{s}_k^{\mathrm{T}} \boldsymbol{y}^k} - \frac{(\boldsymbol{y}^k)^{\mathrm{T}} \boldsymbol{H}_k}{\boldsymbol{s}_k^{\mathrm{T}} \boldsymbol{y}^k}, \quad \boldsymbol{w}_k^2 = \boldsymbol{s}_k,$$

其中 $\mu_k = 1 + \dfrac{(\boldsymbol{y}^k)^{\mathrm{T}} \boldsymbol{H}_k \boldsymbol{y}^k}{\boldsymbol{s}_k^{\mathrm{T}} \boldsymbol{y}^k} \in \mathbb{R}^1$，则得 BFS 算法公式：

$$\begin{cases} \boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \boldsymbol{H}_k \boldsymbol{F}(\boldsymbol{x}^k), \\ \boldsymbol{H}_{k+1} = \boldsymbol{H}_k + \dfrac{\mu_k \boldsymbol{s}_k \boldsymbol{s}_k^{\mathrm{T}} - \boldsymbol{s}_k (\boldsymbol{y}^k)^{\mathrm{T}} \boldsymbol{H}_k - \boldsymbol{H}_k \boldsymbol{y}^k \boldsymbol{s}_k^{\mathrm{T}}}{\boldsymbol{s}_k^{\mathrm{T}} \boldsymbol{y}^k}. \end{cases} \tag{9.7.18}$$

此公式比式(9.7.17)有更好的稳定性，实际计算表明它是拟牛顿法中较为成功的算法.

在一定条件下同样可以证明 DFP 算法(9.7.17)与 BFS 算法(9.7.18)都是超线性收敛的.

9.8 延拓法

9.8.1 延拓法基本思想

在前述的求解方程组(9.1.2)的各种迭代法，基本上都是局部收敛的方法，它要求初始近似 $\boldsymbol{x}^0$ 在方程组的解 $\boldsymbol{x}^*$ 邻域时收敛，否

则不能保证迭代序列收敛，实际计算中要选合适的初始近似 $\boldsymbol{x}^0$ 往往很困难. **延拓法**(continuation)在映射 $\boldsymbol{F}$ 满足一定条件下可从任一点 $\boldsymbol{x}^0$ 出发求得解 $\boldsymbol{x}^*$. 因此，延拓法可看作扩大收敛域的一种有效方法，也是寻找与 $\boldsymbol{x}^*$ 的近似 $\boldsymbol{x}^0$ 的方法，其基本思想就是在所求解的方程组(9.1.2)中引入参数 t，通常取 $t\in[0,1]$，并构造一簇映射 $\boldsymbol{H}$: $D\times[0,1]\subset\mathbb{R}^n\times\mathbb{R}^1\to\mathbb{R}^n$ 代替映射 $\boldsymbol{F}$，使 $\boldsymbol{H}$ 满足条件

$$\boldsymbol{H}(\boldsymbol{x},0)=\boldsymbol{F}_0(\boldsymbol{x}),\quad \boldsymbol{H}(\boldsymbol{x},1)=\boldsymbol{F}(\boldsymbol{x}),\quad \forall\,\boldsymbol{x}\in D,\tag{9.8.1}$$

其中 $\boldsymbol{H}(\boldsymbol{x},0)=\boldsymbol{0}$ 的解 $\boldsymbol{x}^0$ 是已知的，方程 $\boldsymbol{H}(\boldsymbol{x},1)=\boldsymbol{0}$ 是要求解的问题. 如果方程

$$\boldsymbol{H}(\boldsymbol{x},t)=\boldsymbol{0},\quad t\in[0,1]\tag{9.8.2}$$

有解 $\boldsymbol{x}=\boldsymbol{x}(t)$，$\boldsymbol{x}$: $[0,1]\to\mathbb{R}^n$ 连续依赖于 t，当 $t=1$ 时，$\boldsymbol{x}(1)=\boldsymbol{x}^*$ 即为方程(9.1.2)的解，亦即若有连续映射 $\boldsymbol{x}$: $[0,1]\to D\subset\mathbb{R}^n$ 使得

$$\boldsymbol{H}(\boldsymbol{x}(t),t)=\boldsymbol{0},\quad \forall\,t\in[0,1],$$

则 $\boldsymbol{x}=\boldsymbol{x}(t)$ 表示 $\mathbb{R}^n$ 内一条空间曲线，它的一端表示某个给定的点 $\boldsymbol{x}^0$，另一端是 $\boldsymbol{F}(\boldsymbol{x})=\boldsymbol{0}$ 的解 $\boldsymbol{x}^*=\boldsymbol{x}(1)$. 那么，映射 $\boldsymbol{H}$: $D\times[0,1]\subset\mathbb{R}^{n+1}\to\mathbb{R}^n$ 称为**同伦**(homotopy)**映射**. 延拓法就是把求解方程(9.1.2)的问题转化为求同伦方程(9.8.2)的解，故又称为**同伦算法**. 因为延拓法是对原方程嵌入参数 t 而得到的，故又称为**嵌入法**(embedding method).

关于同伦(9.8.2)，解曲线 $\boldsymbol{x}=\boldsymbol{x}(t)$ 的存在性可针对特殊情形给出以下定理.

定理 9.8.1 设映射 $\boldsymbol{F}$: $D\subset\mathbb{R}^n\to\mathbb{R}^n$ 在 D 上连续可导，并假定存在一个开球 $S=S(\boldsymbol{x}^0,r)\subset D$，使对 $\forall\,\boldsymbol{x}\in S$，$\|\boldsymbol{F}'(\boldsymbol{x})^{-1}\|\leqslant\beta$ 成立，其中 $r\geqslant\beta\|\boldsymbol{F}(\boldsymbol{x}^0)\|$，则方程

$$\boldsymbol{F}(\boldsymbol{x})=(1-t)\boldsymbol{F}(\boldsymbol{x}^0),\quad t\in[0,1],\quad \boldsymbol{x}\in S\tag{9.8.3}$$

存在惟一解 $\boldsymbol{x}$: $[0,1]\to S\subset\mathbb{R}^n$，且 $\boldsymbol{x}(t)$ 连续可导并满足微分方程柯西(Cauchy)问题：

$$\begin{cases} \boldsymbol{x}'(t) = -[\boldsymbol{F}'(\boldsymbol{x}(t))]^{-1}\boldsymbol{F}(\boldsymbol{x}^0), \quad \forall t \in [0,1] \\ \boldsymbol{x}(0) = \boldsymbol{x}^0 \end{cases} \tag{9.8.4}$$

此定理表明同伦方程

$$\boldsymbol{H}(\boldsymbol{x},t) = \boldsymbol{F}(\boldsymbol{x}) + (t-1)\boldsymbol{F}(\boldsymbol{x}^0) = 0 \tag{9.8.5}$$

存在惟一解 $\boldsymbol{x}=\boldsymbol{x}(t)$,它也是柯西问题(9.8.5)的解.除式(9.8.5)给出的同伦,满足条件(9.8.1)的同伦还可构造其他形式,常用的有

$$\boldsymbol{H}(\boldsymbol{x},t) = t\boldsymbol{F}(\boldsymbol{x}) + (1-t)\boldsymbol{F}_0(\boldsymbol{x}), \quad \boldsymbol{x} \in D, \quad t \in [0,1], \tag{9.8.6}$$

其中 $\boldsymbol{F}_0$ 是已知映射,且 $\boldsymbol{F}_0(\boldsymbol{x})=\boldsymbol{0}$ 的解 $\boldsymbol{x}^0$ 是给定的.若取 $\boldsymbol{F}_0(\boldsymbol{x})=\boldsymbol{A}(\boldsymbol{x}-\boldsymbol{x}^0)$,$\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 非奇异,则得

$$\boldsymbol{H}(\boldsymbol{x},t) = t\boldsymbol{F}(\boldsymbol{x}) + (1-t)\boldsymbol{A}(\boldsymbol{x}-\boldsymbol{x}^0), \quad \boldsymbol{x} \in D, \quad t \in [0,1]. \tag{9.8.7}$$

当 $\boldsymbol{H}$ 取为同伦(9.8.5)时,即

$$\boldsymbol{H}(\boldsymbol{x},t) = \boldsymbol{F}(\boldsymbol{x}) - (1-t)\boldsymbol{F}(\boldsymbol{x}^0), \quad \boldsymbol{x} \in D, \quad t \in [0,1], \tag{9.8.8}$$

若令 $t=1-\mathrm{e}^{-s}$,可得

$$\boldsymbol{H}(\boldsymbol{x},s) = \boldsymbol{F}(\boldsymbol{x}) - \mathrm{e}^{-s}\boldsymbol{F}(\boldsymbol{x}^0), \quad \boldsymbol{x} \in D, \quad s \in [0,\infty).$$

当 $s=0$,$\boldsymbol{H}(\boldsymbol{x},0)=\boldsymbol{F}(\boldsymbol{x})-\boldsymbol{F}(\boldsymbol{x}^0)=\boldsymbol{0}$ 时有解 $\boldsymbol{x}^0$;当 $s\to\infty$,$\boldsymbol{H}(\boldsymbol{x},s)\to\boldsymbol{F}(\boldsymbol{x})$ 时满足条件(9.8.1),故 $\lim\limits_{s\to\infty}\boldsymbol{x}(s)=\boldsymbol{x}^*$ 是方程 $\boldsymbol{F}(\boldsymbol{x})=\boldsymbol{0}$ 的解.

9.8.2 数值延拓法

假定 $\boldsymbol{H}$: $D\times[0,1]\subset\mathbb{R}^{n+1}\to\mathbb{R}^n$ 是满足式(9.8.1)的已知同伦,若存在一个连续映射 $\boldsymbol{x}$: $[0,1]\to D$,使式(9.8.3)成立,为了求 $\boldsymbol{x}^*=\boldsymbol{x}(1)$,可将区间[0,1]分划为

$$0 = t_0 < t_1 < \cdots < t_N = 1. \tag{9.8.9}$$

用某种迭代法求方程

$$\boldsymbol{H}(\boldsymbol{x},t_i) = \boldsymbol{0} \quad (i = 1,2,\cdots,N) \tag{9.8.10}$$



的解 x^i 时，用第 $i-1$ 个方程的解 x^{i-1} 作初始近似，如果 t_i-t_{i-1} 充分小，则可以期望 x^{i-1} 是 x^i 的一个足够好的近似，从而使迭代法收敛. 由此逐步求得 $H(x,1)=0$ 的解 x^N，即为方程(9.1.2)的近似解，对式(9.8.10)的第 i 个方程求解时，只要用有限步迭代即可，故可取 $m_i\geqslant 1$ 步迭代. 如用牛顿迭代求解方程(9.8.10)，则得数值延拓法序列

$$\begin{cases} x^{i,j+1} = x^{i,j} - [H_x(x^{i,j},t_i)]^{-1}H(x^{i,j},t_i), \\ x^{1,0} = x^0, x^{i+1,0} = x^{i,m_i}, \\ x^{k+1} = x^k - H_x(x^k,1)^{-1}H(x^k,1), \\ x^N = x^{N,0}, \\ j = 0,1,\cdots,m_j-1;\ i = 1,2,\cdots,N-1; \\ k = N,N+1,\cdots \end{cases} \tag{9.8.11}$$

在一定条件下，存在[0,1]的一个分划(9.8.9)和整数序列 $\{m_j\}(j=1,\cdots,N-1)$，使 $\{x^{i,j}\}(i=1,\cdots,N;\ j=0,1,\cdots,m_j)$ 有定义，且 $\lim\limits_{k\to\infty} x^k=x(1)$. 若取 $m_j\equiv 1$，且 $t_i=\dfrac{i}{N}$，H 取为式(9.8.8)时，可得一个大范围收敛的牛顿程序：

$$\begin{cases} x^{k+1} = x^k - F'(x^k)^{-1}\left[F(x^k) - \left(1-\dfrac{k}{N}\right)F(x^0)\right] \\ \qquad (k = 0,1,\cdots,N-1), \\ x^{k+1} = x^k - F'(x^k)^{-1}F(x^k) \quad (k = N,N+1,\cdots). \end{cases} \tag{9.8.12}$$

定理 9.8.2 若 $F: D\subset\mathbb{R}^n\to\mathbb{R}^n$ 有连续导数 $F'(x)$，且 $F'(x)$ 非奇异并满足

$$\| F'(x)^{-1} \| \leqslant \beta < +\infty, \quad \forall x \in D,$$

再假定在 x^0 的邻域 $S=S\{x \mid \| x-x^0 \| \leqslant\beta \| F(x^0) \| \}\subset \mathrm{int}D$，存在 $\gamma>0$，使

$$\| F'(x) - F'(y) \| \leqslant \gamma \| x-y \|, \quad \forall x,y \in S,$$

则以 $\boldsymbol{x}^0$ 为初始值，存在正整数 $N_0 \geqslant 2\beta^2\gamma\|\boldsymbol{F}(\boldsymbol{x}^0)\|$，使当 $N \geqslant N_0$ 时，数值延拓过程(9.8.12)收敛到方程 $\boldsymbol{F}(\boldsymbol{x})=0$ 在 S 中的惟一解 $\boldsymbol{x}^*=\boldsymbol{x}(1)$，且具有平方敛速.

数值延拓法(9.8.12)实际上就是大范围收敛的牛顿法，公式第一式是利用数值延拓法求出 $\boldsymbol{x}(1)$ 的一个足够好的近似 $\boldsymbol{x}^N$，使它进入牛顿收敛域，从而保证牛顿法收敛.

9.8.3 参数微分法

参数微分法是求同伦方程(9.8.2)的另一种数值方法. 若满足同伦方程(9.8.2)的映射 $\boldsymbol{x}: [0,1] \to D$ 可微，且 $\boldsymbol{H}$ 对 $\boldsymbol{x}$ 及 t 有连续偏导数 $\boldsymbol{H}_x$ 及 $\boldsymbol{H}_t$，定义

$$\phi(t) = \boldsymbol{H}(\boldsymbol{x}(t),t), \quad \forall t \in [0,1],$$

则 ϕ 在[0,1]上连续可微，且

$$\phi'(t) = \boldsymbol{H}_x(\boldsymbol{x}(t),t)\boldsymbol{x}'(t) + \boldsymbol{H}_t(\boldsymbol{x}(t),t), \quad \forall t \in [0,1].$$

因 $\boldsymbol{x}=\boldsymbol{x}(t)$ 满足式(9.8.2)，故有 $\phi'(t)=0$，于是 $\boldsymbol{x}(t)$ 满足微分方程

$$\boldsymbol{H}_x(\boldsymbol{x}(t),t)\boldsymbol{x}'(t) = -\boldsymbol{H}_t(\boldsymbol{x}(t),t), \quad \forall t \in [0,1]. \qquad (9.8.13)$$

反之，若 $\boldsymbol{x}: [0,1] \to D$ 是微分方程(9.8.13)的解，并满足 $\boldsymbol{H}(\boldsymbol{x}(0),0)=0$，则由中值定理

$$\|\boldsymbol{H}(\boldsymbol{x}(t),t)\| = \|\phi(t)-\phi(0)\| \leqslant \sup_{0\leqslant s\leqslant t}\|\phi'(s)\| = 0,$$

有

$$\boldsymbol{H}(\boldsymbol{x}(t),t) = 0,$$

即微分方程(9.8.13)及初始条件 $\boldsymbol{H}(\boldsymbol{x}(0),0)=0$ 的解 $\boldsymbol{x}=\boldsymbol{x}(t)$ 给出同伦方程(9.8.2)的一个解. 若 $\boldsymbol{H}_x$ 非奇异，$\boldsymbol{H}(\boldsymbol{x}^0,0)=0$，则求同伦方程(9.8.2)的解 $\boldsymbol{x}=\boldsymbol{x}(t)$ 等价于求微分方程初值问题

$$\begin{cases} \boldsymbol{x}'(t) = -[\boldsymbol{H}_x(\boldsymbol{x},t)]^{-1}\boldsymbol{H}_t(\boldsymbol{x},t), \\ \boldsymbol{x}(0) = \boldsymbol{x}^0 \end{cases} \qquad (9.8.14)$$

的解. 于是，可用常微分方程初值问题的各种数值方法，求问题

(9.8.14)在$t_N=1$的数值解$\boldsymbol{x}^N$. 作为$\boldsymbol{x}=\boldsymbol{x}(1)$的近似解，它也就是方程(9.1.2)的解$\boldsymbol{x}^*$的近似值. 若同伦方程取为(9.8.8)，对应于初值问题(9.8.14)可得

$$\begin{cases}\boldsymbol{x}'(t)=-\boldsymbol{F}'(\boldsymbol{x})^{-1}\boldsymbol{F}(\boldsymbol{x}^0), & \forall t\in[0,1],\\ \boldsymbol{x}(0)=\boldsymbol{x}^0,\end{cases} \tag{9.8.15}$$

若用欧拉(Euler)法求解，步长$h=\dfrac{1}{N}$，则得

$$\boldsymbol{x}^{k+1}=\boldsymbol{x}^k-h\boldsymbol{F}'(\boldsymbol{x}^k)^{-1}\boldsymbol{F}(\boldsymbol{x}^0)\quad(k=0,1,\cdots,N-1). \tag{9.8.16}$$

实际上，当同伦方程有解时就是由式(9.8.15)确定的一条解曲线$\boldsymbol{x}=\boldsymbol{x}(t)$. 当$h$充分小时，由式(9.8.16)确定的$\boldsymbol{x}^k(k=0,1,\cdots,N)$应当近似这条曲线，$\boldsymbol{x}^N$可望与$\boldsymbol{x}^*=\boldsymbol{x}(1)$充分靠近，且由$\boldsymbol{x}^N$出发，用牛顿法迭代将收敛于$\boldsymbol{x}^*$. 图9.3表示连接$\boldsymbol{x}^0$与$\boldsymbol{x}^*$的解曲线$\boldsymbol{x}=\boldsymbol{x}(t)$及相应数值解$\boldsymbol{x}^k$.

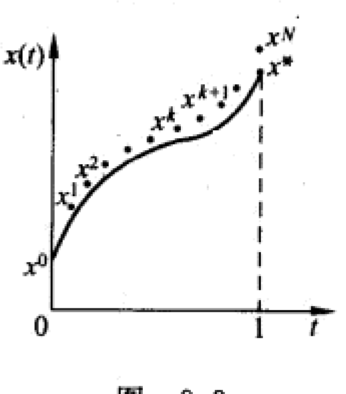


图 9.3

当$\boldsymbol{F}:\mathbb{R}^n\to\mathbb{R}^n$在$\mathbb{R}^n$上二次连续可微并且$\|\boldsymbol{F}'(\boldsymbol{x})^{-1}\|\leqslant\beta$时，对任何$\boldsymbol{x}^0\in\mathbb{R}^n$，存在一个$N_0\geqslant1$，使得$N\geqslant N_0$，则由式(9.8.16)求得的$\boldsymbol{x}^N$为初始近似，牛顿迭代$\boldsymbol{x}^{k+1}=\boldsymbol{x}^k-\boldsymbol{F}'(\boldsymbol{x}^k)^{-1}\times\boldsymbol{F}(\boldsymbol{x}^k)(k=0,1,\cdots)$生成的序列收敛于$\boldsymbol{F}(\boldsymbol{x})=\boldsymbol{0}$的惟一解.

实际使用参数微分法时，还可用精度较高的数值方法求初值问题(9.8.15)的解. 例如下列中点求积公式：

$$\begin{cases}\boldsymbol{x}^1=\boldsymbol{x}^0-\dfrac{1}{N}[\boldsymbol{F}'(\boldsymbol{x}^0)]^{-1}\boldsymbol{F}(\boldsymbol{x}^0),\\ \boldsymbol{x}^{k+\frac{1}{2}}=\boldsymbol{x}^k+\dfrac{1}{2}(\boldsymbol{x}^k-\boldsymbol{x}^{k-1}), & k=1,\cdots,N-1\\ \boldsymbol{x}^{k+1}=\boldsymbol{x}^k-\dfrac{1}{N}[\boldsymbol{F}'(\boldsymbol{x}^{k+\frac{1}{2}})]^{-1}\boldsymbol{F}(\boldsymbol{x}^0),\end{cases} \tag{9.8.17}$$

具有二阶精度，其计算工作量与欧拉法相当，可用于实际计算.

例 9.8.3 用参数微分法求下列方程

$$\boldsymbol{F}(\boldsymbol{x}) = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} = \begin{pmatrix} x_1^2 - x_2 + 1 \\ x_1 - \cos\left(\frac{\pi}{2}x_2\right) \end{pmatrix} = \boldsymbol{0}$$

的近似解，给定初始近似 $\boldsymbol{x}^0 = (1,0)^{\mathrm{T}}$.

解 此方程由给定 $\boldsymbol{x}^0$ 出发，直接用牛顿法(9.3.2)迭代不收敛；若用牛顿下山法(9.3.19)，要迭代 107 次才收敛到 $\boldsymbol{x}^* = (0, 1)^{\mathrm{T}}$；若用公式(9.8.17)，取 $N = 8$ 可求得 $\boldsymbol{x}^N = (0.095552027, 0.97843313)^{\mathrm{T}}$，从 $\boldsymbol{x}^N$ 出发用牛顿迭代(9.3.1)做三次迭代，得

$$\boldsymbol{x}^* \approx (0.000000049, 0.999999963)^{\mathrm{T}}.$$

9.9 单纯形算法

单纯形算法(simplicial algorithm)是 20 世纪 60 年代后期出现的计算不动点和求非线性方程组零点的新算法，是拓扑学与计算数学结合的产物. 算法的理论基础是 Sperner 引理. 1967 年，H. Scarf 首先提出了映射不动点的构造性逼近方法，体现了单纯形算法的思想，Cohen 给出了 Sperner 引理的构造性证明，随后 Kuhn、Eaves 等人又提出了几种新算法. 这些方法不但在理论上有重要意义，而且提供了一类具有大范围收敛的可行算法.

9.9.1 三角剖分与算法基本思想

定义 9.9.1 设 $\boldsymbol{u}^0, \boldsymbol{u}^1, \cdots, \boldsymbol{u}^m \in \mathbb{R}^n$ 在一般位置上，所有满足

$$\boldsymbol{x} = \sum_{j=0}^{m} \lambda_j \boldsymbol{u}^j, \quad \sum_{j=0}^{m} \lambda_j = 1, \quad \lambda_j \geqslant 0 \quad (j = 0,1,\cdots,m) \tag{9.9.1}$$

的点集合 σ，称为一个 $\boldsymbol{m}$ **维单纯形**(m-simplex)，记作 $\sigma = (\boldsymbol{u}^0,$

$\boldsymbol{u}^1,\cdots,\boldsymbol{u}^m)$. 点 $\boldsymbol{u}^0,\boldsymbol{u}^1,\cdots,\boldsymbol{u}^m$ 称为 σ 的**顶点**，记作

$$\mathrm{Vert}(\sigma)=\{\boldsymbol{u}^0,\boldsymbol{u}^1,\cdots,\boldsymbol{u}^m\}.$$

根据定义，当 $m\leqslant n$ 时，σ 是一个 m 维凸多面体. 当 $m=1$ 时，是直线段，当 $m=2$ 时，是平面上三角形，当 $m=3$ 时，是空间中的四面体. 如果另有一个 k 维单纯形 τ，满足

$$\mathrm{Vert}(\tau)\leqslant\mathrm{Vert}(\sigma),$$

则 τ 称为 σ 的一个 **k 维面**. 由于 $k\leqslant m$，所以，σ 的零维面即 σ 的 $m+1$ 个顶点，σ 的一维面即多面体的棱，m 维面是 σ 本身. σ 的 $m-1$ 维面称为 σ 的**真面**. 如果 τ 的顶点中只比 σ 的顶点缺少 u^j，记作 $\tau=\mathrm{opp}(\sigma,u^j)$，称这个真面 τ 为 σ 中 u^j 的**对面**. 所有 m 个真面的并集称为 σ 的**边界**，记作

$$\partial(\sigma)=\bigcup_{j=0}^{m}\mathrm{opp}(\sigma,u^j).$$

定义 9.9.2 $\mathbb{R}^n$ 中两个 m 维单纯形 σ_1 和 σ_2 称为**规则相联的**，如果 $\sigma_1\neq\sigma_2$，且 $\sigma_1\cap\sigma_2$ 是两者的一个 $m-1$ 维公共面，如果有 N 个 m 维单纯形 $\sigma_1,\cdots,\sigma_N$ 两两之间规则相联，则其并集

$$D=\bigcup_{i=1}^{N}\sigma_i$$

称为$\mathbb{R}^n$中的一个 m **维复形**. σ_i 的集合

$$G=\{\sigma_i\mid i=1,\cdots,N\}$$

称为 D 的一个**单纯形剖分或三角剖分**.

单纯形剖分 G 的格长记为 $\mathrm{mesh}(G)$，

$$\mathrm{mesh}(G)=\max_{\sigma\in G}\{\mathrm{diam}(\sigma)\},$$

其中 $\mathrm{diam}(\sigma)=\max\limits_{x,y\in\sigma}\|x-y\|$ 为单纯形 σ 的直径.

G 中所有单纯形的 k 维面集合称为 G 的 **k 维骨架**，记作 G^k，于是 $G^m=G$，G^0 为 G 的顶点集合. 关于 D 的单纯形剖分 G，有以下的直观性质：

(1) 若 $\sigma_1,\sigma_2\in G$ 且 $\sigma_1\cap\sigma_2\neq\varnothing$（空集），则

$$\tau = \sigma_1 \cap \sigma_2 \in G^{m-1}.$$

(2) 若 $\tau \in G^{m-1}$，那么，$\tau \subset \partial D$，τ 只是 G 的一个单纯形界面，或者 $\tau \not\subset \partial D$，$\tau$ 恰好是 G 中两个单纯形的公共界面.

(3) 存在 G 的格长 mesh(G)任意小的剖分.

设 $\boldsymbol{F}$: $D \subset \mathbb{R}^n \to \mathbb{R}^n$，$G$ 为 D 上一个 n 维单纯形剖分，记 $N_0 = \{0,1,\cdots,n\}$，则任何 $l: G^0 \to N_0$ 都可称为一个**整数标号**. 例如，可由 $\boldsymbol{F}$ 定义 l，对任何 $u \in G^0$，定义

$$l(u) = \begin{cases} i, & \text{若 } f_i(u) > 0, \quad \text{且 } f_j(u) \leqslant 0, \quad \forall j < i, \\ 0, & \text{若 } f_j(u) \leqslant 0, \quad j = 1,2,\cdots,n, \end{cases} \tag{9.9.2}$$

这里 f_j 是 $\boldsymbol{F}$ 的第 j 个分量，即 $\boldsymbol{F} = (f_1,\cdots,f_n)^{\mathrm{T}}$.

G 中一个具有$\{0,1,\cdots,n\}$标号的单纯形 σ，称为**全标号单纯形**. G 中具有$\{0,1,\cdots,n-1\}$标号，而无 n 标号的 n 维单纯形称为**几乎全标号单纯形**. 按式(9.9.2)的标号定义，有以下的定理.

定理 9.9.3 设 $\boldsymbol{F}$: $D \subset \mathbb{R}^n \to \mathbb{R}^n$，$D$ 为有界闭凸集，若 $\boldsymbol{F}$ 在 D 上连续(即对任给的 $\epsilon > 0$，$\exists \delta > 0$，对任何 $\boldsymbol{x}, \boldsymbol{y} \in D$，只要 $\|\boldsymbol{x} - \boldsymbol{y}\| \leqslant \delta$，就有 $\|\boldsymbol{F}(\boldsymbol{x}) - \boldsymbol{F}(\boldsymbol{y})\| \leqslant \epsilon$)，若 D 的单纯形剖分 G 的格长 mesh$(G) \leqslant \delta$，且 σ 为 G 中一个全标号单纯形，则对任何 $\boldsymbol{x} \in \sigma$，有 $\|\boldsymbol{F}(\boldsymbol{x})\| \leqslant \epsilon$.

根据定理，在有界闭集 D 上求方程(9.1.2)的解，只要在 D 上建立一串 mesh$(G_k) \to 0$ 的单纯形剖分 $G_k(k=0,1,\cdots)$，在每个 G_k 上又有一个按式(9.9.2)定义的全标号单纯形 σ_k，则$\{\sigma_k\}$当 $k \to \infty$ 时有极限点 $\boldsymbol{x}^*$，由定理 9.9.3 可知这个极限点 $\boldsymbol{x}^*$ 就是方程 $\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{0}$ 的解，这就是单纯形算法的基本思想，算法包括三步：

(1) 对 D 进行三角剖分 $G = \{\sigma_i \mid i = 1,2,\cdots,N\}$；

(2) 对 G 的顶点集合 G^0 进行标号；

(3) 用一种有规则的搜索方法求 G 的全标号单纯形 σ.

例 9.9.4 用单纯形算法求方程

$$F(x,y)=\begin{pmatrix}x+y-1.5\\ x^2-y^2-0.15\end{pmatrix}=0$$

在(0,0),(1,0),(1,1)围成的区域 D 上的解.

解 第一步,先对域 D 进行三角剖分 G,如图 9.4 所示,这里用的是$\frac{1}{4}K_1$ 剖分.

第二步,对各顶点标号,这里 $f_1=x+y-1.5$, $f_2=x^2-y^2-0.15$;按公式(9.9.2)计算得到 G 的各顶点标号,均标在图 9.4 上.

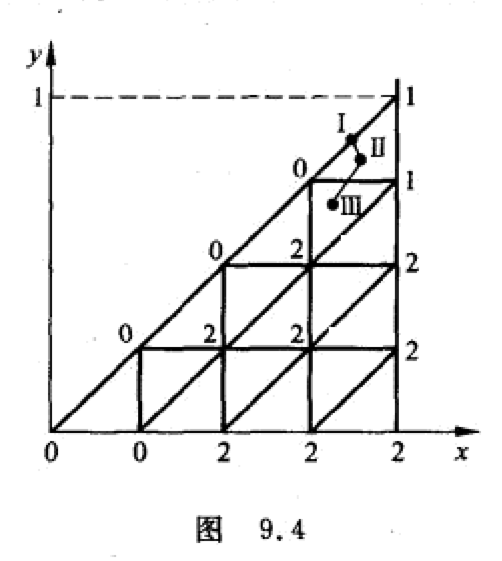


图 9.4

第三步,搜索全标号单纯形,这里用的方法是从 D 的边界一维全标号单形Ⅰ开始进入二维几乎全标号单形Ⅱ,再进入全标号单形Ⅲ,即为所求.若取Ⅲ的重心坐标 $x=0.875$, $y=0.625$ 作为方程的近似解,此时 $f_1=0$, $f_2=0.225$,此题的精确解为 $x^*=0.8$, $y^*=0.7$.为提高精度可把剖分再缩小 1/2,找到格长更小的全标号单纯形,其重心坐标为 $x=0.8125$, $y=0.6875$,更接近精确解.若精度不够,还可缩小剖分格长 mesh(G),直到满足精度要求为止.

9.9.2 同伦算法

尽管单纯形算法具有只要求 F 连续，且收敛是大范围的优点，但是，有两个严重弱点，一是当步长选定后，整个计算都要按此步长进行. 二是搜索全标号单纯形必须从边上开始，这样步长取大了，精度很差；步长取小了，路径上节点很多，计算量太大，使得实际计算几乎不可能. 因此，单纯形算法只能作为求初始近似的方法. 针对单纯形算法的弱点，Merrill 于 1972 年提出了**重启动**(restart)**法**，Eaves 提出了**同伦**(homotopy)**算法**，又称**连续变形算法**(continuous deformation algorithm). 这两种方法从理论分析和计算实践都表明是可行的，此后出现的新算法均以此为基础.

定义同伦 $\boldsymbol{H}$: $D\times[0,1]\subset\mathbb{R}^{n+1}\to\mathbb{R}^n$ 为

$$\boldsymbol{H}(\boldsymbol{x},t)=\begin{cases}2t\boldsymbol{F}(\boldsymbol{x})+(1-2t)\boldsymbol{F}_0(\boldsymbol{x}), & 0\leqslant t\leqslant 1/2,\\ \boldsymbol{F}(\boldsymbol{x}), & 1/2\leqslant t\leqslant 1,\end{cases}\tag{9.9.3}$$

满足条件(9.8.1). 对 $D\times[0,1]$ 进行三角剖分 G，它具有以下性质：对 $D\times[1-2^{-k},1-2^{-k-1}]$ 的三角剖分 $G_k(k=0,1,\cdots)$ 为 G 的子集，且当 $k\to\infty$，$\mathrm{mesh}(G_k)\to 0$，$n=1$ 时，这种剖分如图 9.5 所示.

定义 9.9.5 设 $\boldsymbol{F}$: $D\in\mathbb{R}^n\to\mathbb{R}^n$，对凸集 D 有三角剖分 G，$\sigma=\langle\boldsymbol{v}^0,\cdots,\boldsymbol{v}^n\rangle\in G$，对任何 $\boldsymbol{x}\in\sigma$，若 $\boldsymbol{x}=\sum_{i=0}^{n}\lambda_i\boldsymbol{v}^i$，$\lambda_i\geqslant 0$ 且 $\sum_{i=0}^{n}\lambda_i=1$，则定义 $\boldsymbol{P}$: $D\subset\mathbb{R}^n\to\mathbb{R}^n$ 为

$$\boldsymbol{P}(\boldsymbol{x})=\sum_{i=0}^{n}\lambda_i\boldsymbol{F}(\boldsymbol{v}_i)$$

称为基于 G 的一个**分片线性逼近**(piecewise iinear approximation). $\boldsymbol{P}(\boldsymbol{x})$ 可作为 $\boldsymbol{F}$ 的一个向量标号函数.

若 $\boldsymbol{P}(\boldsymbol{x}^*)=\boldsymbol{0}$，$\boldsymbol{x}^*\in D$，则称 $\boldsymbol{x}^*$ 为 $\boldsymbol{F}$ 在三角剖分 G 下的一个近似解. 若 $\boldsymbol{x}^*\in\sigma\in G$ 则 σ 称为一个 n 维**完全**(complete)**单纯形**.

定理 9.9.6 设 $\boldsymbol{F}: D\subset\mathbb{R}^n\to\mathbb{R}^n$ 连续可微，且 $\boldsymbol{F}'(\boldsymbol{x})$ 在凸集 D 上利普希茨连续，即 $\exists\gamma>0$，使

$$\|\boldsymbol{F}'(\boldsymbol{x})-\boldsymbol{F}'(\boldsymbol{y})\|\leqslant\gamma\|\boldsymbol{x}-\boldsymbol{y}\|,\quad\forall\boldsymbol{x},\boldsymbol{y}\in D,$$

若 $\boldsymbol{x}^*$ 是 $\boldsymbol{F}$ 在三角剖分 G 下的近似解，$\mathrm{mesh}(G)\leqslant\delta$，则有

$$\|\boldsymbol{F}(\boldsymbol{x})\|\leqslant\gamma\delta^2,$$

这里 $\|\cdot\|$ 均为 l_∞ 范数.

根据定理只要找到完全单纯形 σ，也就找到了 $\boldsymbol{F}(\boldsymbol{x})=\boldsymbol{0}$ 的近似解，它比全标号单纯形具有更高的精度，问题是如何找到完全单纯形. 当 $\boldsymbol{F}: D\subset\mathbb{R}^n\to\mathbb{R}^n$，$\sigma=\langle\boldsymbol{v}^0,\cdots,\boldsymbol{v}^n\rangle\in G$ 时，称 $(n+1)$ 阶矩阵

$$\boldsymbol{L}_\sigma=\begin{pmatrix}1&1&\cdots&1\\\boldsymbol{F}(\boldsymbol{v}^0)&\boldsymbol{F}(\boldsymbol{v}^1)&\cdots&\boldsymbol{F}(\boldsymbol{v}^n)\end{pmatrix}$$

为 σ 的**标号矩阵**(label matrix).

σ 为 n 维完全单纯形的充分必要条件是

$$\boldsymbol{L}_\sigma\boldsymbol{w}=\boldsymbol{e}^1,\boldsymbol{w}\geqslant\boldsymbol{0},\boldsymbol{w}\in\mathbb{R}^{n+1},\boldsymbol{e}^1=(1,0,\cdots,0)^{\mathrm{T}}\in\mathbb{R}^{n+1}$$

有解.

设同伦 $\boldsymbol{H}: D\times[0,1]\subset\mathbb{R}^{n+1}\to\mathbb{R}^n$，对 $D\times[1-2^{-k},1-2^{-k-1}]$ 的三角剖分 $G_k(k=0,1,\cdots)$，若

$$\tau=\langle\boldsymbol{v}^0,\cdots,\boldsymbol{v}^n,\boldsymbol{v}^{n+1}\rangle\in G_k\subset G,$$

同样称 $(n+1)\times(n+2)$ 矩阵

$$\boldsymbol{L}_\tau=\begin{pmatrix}1&1&\cdots&1\\\boldsymbol{H}(\boldsymbol{v}^0)&\boldsymbol{H}(\boldsymbol{v}^1)&\cdots&\boldsymbol{H}(\boldsymbol{v}^{n+1})\end{pmatrix}$$

为 τ 的标号矩阵. 当

$$\boldsymbol{L}_\tau\boldsymbol{w}=\boldsymbol{e}^1,\boldsymbol{w}\geqslant\boldsymbol{0},\boldsymbol{w}\in\mathbb{R}^{n+2},\boldsymbol{e}^1=(1,0,\cdots,0)^{\mathrm{T}}\in\mathbb{R}^{n+1}\quad(9.9.4)$$

有解时，称 τ 为一个 $(n+1)$ 维**几乎完全单纯形**. 根据线性规划基本定理，当矩阵 $\boldsymbol{L}_\tau$ 的秩为 $n+1$ 时，若式(9.9.4)有解，则它恰好有两个基本可行解. 这就说明，G 中一个 $n+1$ 维几乎完全单纯形 τ，当 $\boldsymbol{L}_\tau$ 的秩为 $n+1$ 时恰有两个 n 维面是完全单纯形，定义这样的两个 n 维单纯形是相邻的. 另一方面，对不在 $D\times\{1-2^{-k}\}(k=0,$

1,⋯)上的 n 维完全单纯形,它一定正好属于两个 G 中的 $n+1$ 维几乎完全单纯形. 于是,当 $\boldsymbol{x}^0$ 为 $D\times\{0\}$ 上的一个 n 维完全单纯形内点时,由此开始形成一个惟一的完全单纯形轮回路径,实际上就是同伦方程 $\boldsymbol{H}(\boldsymbol{x},t)=\boldsymbol{0}$,对 $t\in[0,1]$的解的轨迹,当 $t\to 1$(即 $k\to\infty$),解 $\boldsymbol{x}(t)$就趋向 $\boldsymbol{F}(\boldsymbol{x})=\boldsymbol{0}$ 的解.

设 $\sigma_0=(\boldsymbol{v}^0,\cdots,\boldsymbol{v}^n)$是三角剖分 G_0 包含$(\boldsymbol{x}^0,0)^{\mathrm{T}}$ 的一个 n 维面,σ_0 是 n 维完全单纯形,于是同伦算法由此开始,其算法步骤如下:

(1) 由 σ_0 求 $\tau_0=\langle\boldsymbol{v}^0,\cdots,\boldsymbol{v}^n,\boldsymbol{v}^{n+1}\rangle\in G_0$, $0\to p$.

(2) 设 σ_p 是 τ_p 的完全单纯形 n 维面,由 τ_p 求另一个完全单纯形 n 维面 σ_{p+1}.

(3) 若 $\sigma_{p+1}\in D\times\{1-2^{-k}\}$,且 $2^{-k}\leqslant\varepsilon$(给定精度),则转第(4)步;否则,由 σ_{p+1}求 $\tau_{p+1}\neq\tau_p$(τ_{p+1}与 τ_p 规则相联),$p+1\to p$ 并转第(2)步.

(4) $\sigma_{p+1}=\langle\boldsymbol{u}^0,\cdots,\boldsymbol{u}^n\rangle=\langle(\boldsymbol{x}^0,1-2^{-k})^{\mathrm{T}},\cdots,(\boldsymbol{x}^n,1-2^{-k})^{\mathrm{T}}\rangle$是一个 n 维完全单纯形,存在权 $w_0,\cdots,w_n$ 使 $\boldsymbol{u}^*=\sum\limits_{i=0}^{n}w_i\boldsymbol{u}^i$ 是 $\boldsymbol{H}(\boldsymbol{x},1-2^{-k})$ 的一个零点,而 $\boldsymbol{x}^*=\sum\limits_{i=0}^{n}w_i\boldsymbol{x}^i$ 是 $f(\boldsymbol{x})$的一个零点.

例 9.9.7 令 $n=1$, $F(x)\triangleq x^3+2$, $F_0(x)=x-1/2$, $D\times[0,1]$的三角剖分 G 及 $H^{-1}(0)$的轨迹如图 9.5 所示.

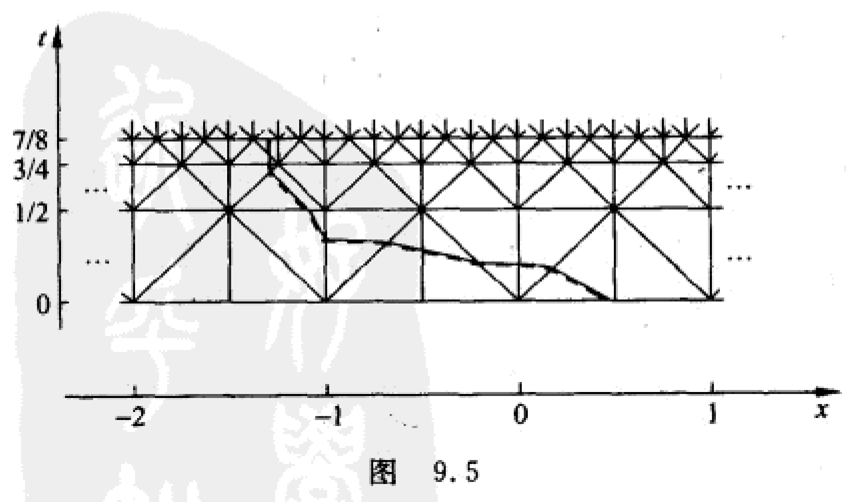


图 9.5

9.10 区间迭代法

求解非线性方程组

$$f(x) = 0 \quad (f: D \subset \mathbb{R}^n \to \mathbb{R}^n) \tag{9.10.1}$$

的区间迭代法，首先是由 Moore 于 1966 年提出的，它以区间为变量，将区间映到区间. Moore 给出的区间牛顿算子是

$$N(X) = x - F'(X)^{-1} f(x), \quad \forall x \in X, \tag{9.10.2}$$

这里 $X \subset D \subset \mathbb{R}^n$ 为 $\mathbb{R}^n$ 中的区间向量，$X = (X_1, \cdots, X_n)^{\mathrm{T}}$ 是一个 n 维长方体，$X_i (i = 1, 2, \cdots, n)$ 都是区间. $F'(X)$ 是 $f'(x)$ 的包含区间扩展，即对任何 $x \in X$，有 $f'(x) \in F'(X)$. $F'(X)^{-1}$ 就是一个区间矩阵的逆. 因此，式(9.10.2)给出的算子 $N(X)$ 仍然是一个 n 维区间向量，并有以下性质：

(1) 若 $x^* \in X$ 是方程组(9.10.1)的解，则 $x^* \in N(X)$；

(2) 若 $X \cap N(X) = \varnothing$（空集），则方程组(9.10.1)在 X 中无解；

(3) 若 $N(X) \subset X$ 则方程组(9.10.1)在 X 中有解 x^*，且 $x^* \in N(x)$；若 $W(N(X)) < W(X)$（$W(X)$ 表示区间 X 的宽度，当 $X = (X_1, \cdots, X_n)^{\mathrm{T}}$，宽度 $W(X) = (W(X_1), \cdots, W(X_n))^{\mathrm{T}}$），则方程组(9.10.1)在 X 中的解 x^* 是惟一的.

根据 $N(X)$ 的性质，可构造区间牛顿迭代

$$X^{k+1} = X^k \cap N(X^k) \quad (k = 0, 1, \cdots). \tag{9.10.3}$$

若对某个 k 有 $N(X^k) \subset X^k$，且 $W(N(X^k)) < W(X^k)$，则 $\lim\limits_{k\to\infty} X^k = x^*$ 为方程(9.10.1)的解；若 $X^k \cap N(X^k) = \varnothing$，则方程组(9.10.1)在 X^0 中无解. 这表明区间牛顿法每步迭代都可检验方程(9.10.1)解的存在惟一性，这是点迭代法所不能解决的，也是区间迭代法的主要优点. 但是，求 $N(X)$ 要计算区间矩阵 $F'(X)$ 的逆，要按区间运算规划求 $f'(x)$ 的区间扩展 $F'(X)$，再求逆，因此，解区间线性方

程组的工作量很大，故很不实用. 为减少计算工作量，避免求逆，Krawczyk 在算子 $N(X)$基础上引进新的区间算子

$$K(y,X) = y - Yf(y) + [I - YF'(X)](X - y), \tag{9.10.4}$$

其中 $y \in X$ 是 X 中任一点，Y 为非奇的 n 阶矩阵，即 $Y \in \mathbb{R}^{n\times n}$，$I$ 为 n 阶单位矩阵. $F'(X)$仍为 $f'(x)$的区间扩展，称 $K(y,X)$为 Krawczyk **算子**. 区间向量 $X=[\underline{x},\bar{x}]=\{x\in\mathbb{R}^n \mid \underline{x}\leqslant x\leqslant\bar{x}\}$是$\mathbb{R}^n$中的紧凸集，由不动点定理可知，若 $\bar{N}(x)=x-Yf(x)$映区间 X 于自身，即 $\bar{N}(X)\subset X$，则存在 $x^*\in X$ 为 $\bar{N}(x)$的不动点，于是 $f(x^*)=0$，而对任何 $x,y\in X$，有

$$\bar{N}(x) - \bar{N}(y) = \bar{N}'(\xi)(x - y), \forall\, \xi \in X, \bar{N}'(x) = I - Yf'(x),$$

故

$$\bar{N}(x) \in K(y,X) = \bar{N}(y) + \bar{N}'(X)(X - y).$$

根据 Krawczyk 算子的这一性质，可以得到检验方程(9.10.1)解存在惟一性的定理.

定理 9.10.1 设 $f: X=[\underline{x},\bar{x}]\subset\mathbb{R}^n\to\mathbb{R}^n$在 X 中连续可微，且 f, f'包含区间扩展 F 及 F'，$K(y,X)$是由(9.10.4)定义的 Krawczyk 算子，则

(1) 方程(9.10.1)在 X 中的解都在$K(y,X)$中；

(2) 若 $X\cap K(y,X)=\varnothing$，则方程(9.10.1)在 X 中无解，若 $K(y,X)\subset X$，则方程(9.10.1)在 X 中至少有一个解；

(3) 若 $K(y,X)\subset X$，且 $W(K(y,X))<W(X)$，则方程(9.10.1)在X 中有惟一解 x^*，且以 $x^0\in X$ 为初始近似的迭代序列

$$x^{(k+1)} = x^{(k)} - Yf(x^{(k)}) \quad (k = 0,1,\cdots)$$

收敛到 x^*，且

$$| X^{(k)} - x^* | \leqslant \beta^k W(X),$$

其中

$$\beta = \max_{1 \leqslant i \leqslant n} \{ W(K(y, X_i)) / W(X_i) \} < 1.$$

此定理也称为 Moore 检验，它可通过下列区间迭代法验检解的存在惟一性.

算法 9.10.2

取 $X^0 = X = [\underline{x}, \overline{x}]$，$y^0 = m(X)$ $\left(m(X) = \dfrac{\underline{x} + \overline{x}}{2} \text{称为区间} X \text{的中点} \right)$. 对 $k=0,1,\cdots$，计算

$$K(y^k, X^k) = y^k - Y_k f(y^k) + [I - Y_k F'(X^k)](X^k - y^k),$$

其中 $y^k = m(X^k)$，$Y_k = [m(F'(X^k))]^{-1}$. 若 $X^k \cap K(y^k, X^k) = \varnothing$，则停止计算，方程在 X 中无解；否则令

$$X^{k+1} = X^k \cap K(y^k, X^k),$$

若 $|W(X^{k+1})| \leqslant \varepsilon$（给定精度），则停止计算，并取 $m(X^{k+1})$ 作为 x^* 近似；否则 $k+1 \to k$ 重新计算.

只要存在某个 l，使 $K(y^l, X^l) \subset X^l$，且 $W(K(y^l, X^l)) < W(X^l)$ 成立，则

$$\lim_{k \to \infty} X^k = x^*.$$

算法 9.10.2 以及 Moore 检验定理 9.10.1 体现了区间迭代法的特点，计算量也比区间牛顿法(9.10.3)大为减少，它可直接用来判断方程组(9.10.1)在区间 $X = [\underline{x}, \overline{x}]$ 中是否有解以及解是否惟一，如解存在惟一还可得到近似解的误差估计.

1980 年，Nickel 给出一种球形牛顿法，它是区间迭代法的另一个杰出成就. 在 $\mathbb{R}^n$ 中的 n 维球 $A = \{ \boldsymbol{x} \in \mathbb{R}^n \mid \| \boldsymbol{x} - \boldsymbol{a} \| \leqslant r \}$ 可表示为 $A = \langle \boldsymbol{a}, \boldsymbol{r} \rangle$，$\boldsymbol{a} \in \mathbb{R}^n$ 为中心，记作 $\mathrm{mid} A = \boldsymbol{a}$；$r$ 为半径，记作 $\mathrm{rad} A = r$. 对 $\boldsymbol{x} \in \mathbb{R}^n$，定义正则球算子

$$L\boldsymbol{x} = \langle \boldsymbol{\Lambda x}, \lambda \| \boldsymbol{\Lambda x} \| \rangle$$

其中 $\boldsymbol{\Lambda}$ 为 n 阶非奇异矩阵，$\lambda \in (0,1)$，记 $L = \langle\!\langle \boldsymbol{\Lambda}, \lambda \rangle\!\rangle$. 设 $\boldsymbol{f}: D \subset \mathbb{R}^n \to \mathbb{R}^n$，对每个形如

若 $\overline{\boldsymbol{X}}^{k+1}\cap\boldsymbol{X}^0=\varnothing$，则算法停止. 若 $\boldsymbol{X}^{k+1}\cap\overline{\boldsymbol{X}}^0\neq\varnothing$，且 $\boldsymbol{X}^{k+1}=\mathrm{mid}\overline{\boldsymbol{X}}^{k+1}$. 则当 $\bar{\boldsymbol{x}}^{k+1}\in\boldsymbol{X}^0$ 定义 $\boldsymbol{X}^{k+1}=\overline{\boldsymbol{X}}^{k+1}$；否则定义 $\boldsymbol{X}^{k+1}=\langle\boldsymbol{X}^0\cap\overline{\boldsymbol{X}}^{k+1}\rangle$. 这样定义的球形牛顿法是全局收敛的. 球形牛顿迭代同样具有区间迭代的特点，但计算交球仍然较复杂.

9.11 并行多分裂方法

多分裂方法是为多处理机设计的适合并行计算的有效算法，这类方法自 20 世纪 80 年代提出后获得很大发展，在求解大型线性问题与非线性问题中已被广泛应用. 特别是多处理机在科学计算中已较普遍使用的今天，此类算法更显出它的重要性，而且它不是把已有的串行算法并行化，而是将大规模问题“分而治之”转化为若干小规模问题，使计算效率大为提高，算法基础是线性多分裂.

9.11.1 线性多分裂方法

给定线性方程组

$$\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b},\tag{9.11.1}$$

$\boldsymbol{A}\in\mathbb{R}^{n\times n}$，$\boldsymbol{x},\boldsymbol{b}\in\mathbb{R}^n$，$\boldsymbol{A}$ 可分裂为

$$\boldsymbol{A}=\boldsymbol{M}-\boldsymbol{N},\tag{9.11.2}$$

其中 $\boldsymbol{M},\boldsymbol{N}\in\mathbb{R}^{n\times n}$，$\boldsymbol{M}$ 非奇且容易求逆，迭代法

$$\boldsymbol{x}^{k+1}=\boldsymbol{M}^{-1}\boldsymbol{N}\boldsymbol{x}^k+\boldsymbol{M}^{-1}\boldsymbol{b}\quad(k=0,1,\cdots)\tag{9.11.3}$$

取决于不同的分裂，而线性多分裂就是上述思想的推广，对 $\boldsymbol{A}$ 作多种形如式(9.11.2)的分裂，并分别确定形如式(9.11.3)的相应迭代，总体迭代定义为各种迭代进程的线性组合. 具体地给出下面的定义.

定义 9.11.1 设矩阵 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$，若存在矩阵 $\boldsymbol{M}_l,\boldsymbol{N}_l,\boldsymbol{E}_l\in\mathbb{R}^{n\times n}(l=1,2,\cdots,L)$，满足

(1) $\boldsymbol{A}=\boldsymbol{M}_l-\boldsymbol{N}_l \quad (l=1,2,\cdots,L)$；

(2) $\boldsymbol{M}_l(l=1,2,\cdots,L)$非奇异；

(3) $\boldsymbol{E}_l(l=1,2,\cdots,L)$为对角元素非负的对角矩阵，且

$$\sum_{l=1}^{L}\boldsymbol{E}_l=\boldsymbol{I} \quad (n\text{ 阶单位矩阵}),$$

则称三重集合$\{(\boldsymbol{M}_l,\boldsymbol{N}_l,\boldsymbol{E}_l)\mid l=1,2,\cdots,L\}$是$\boldsymbol{A}$的一个多分裂.对应的迭代法

$$\boldsymbol{x}^{k+1}=\sum_{l=1}^{L}\boldsymbol{E}_l\boldsymbol{M}_l^{-1}(\boldsymbol{N}_l\boldsymbol{x}^k+\boldsymbol{b}) \quad (k=0,1,\cdots) \tag{9.11.4}$$

称为解线性方程组(9.11.1)的**线性多分裂方法**.

显然式(9.11.4)可写成

$$\boldsymbol{x}^{k+1}=\boldsymbol{G}\boldsymbol{x}^k+\boldsymbol{f} \quad (k=0,1,\cdots), \tag{9.11.5}$$

其中

$$\boldsymbol{G}=\sum_{l=1}^{L}\boldsymbol{E}_l\boldsymbol{M}_l^{-1}\boldsymbol{N}_l,\boldsymbol{f}=\sum_{l=1}^{L}\boldsymbol{E}_l\boldsymbol{M}_l^{-1}\boldsymbol{b}. \tag{9.11.6}$$

迭代法(9.11.4)具有内在的并行性，因为对每个不同的l，$\boldsymbol{E}_l\boldsymbol{M}_l^{-1}(\boldsymbol{N}_l\boldsymbol{x}^k+\boldsymbol{b})$的计算是独立的可以并行执行，当$\boldsymbol{E}_l$的第$i$个对角元素为零时，$\boldsymbol{M}_l^{-1}(\boldsymbol{N}_l\boldsymbol{x}^k+\boldsymbol{b})$的第$i$个分量无需计算，因此，只要多分裂$\boldsymbol{M}_l-\boldsymbol{N}_l$取得适当，就可利用无需计算的若干分量节省计算时间.对迭代法(9.11.5)收敛的充分必要条件是$\rho(\boldsymbol{G})<1$，为给出迭代收敛的充分条件，需用以下定义.

定义 9.11.2 设矩阵$\boldsymbol{A}\in\mathbb{R}^{n\times n}$分裂为$\boldsymbol{A}=\boldsymbol{M}-\boldsymbol{N}$.若$\boldsymbol{M}^{-1}\geqslant\boldsymbol{0}$，$\boldsymbol{N}\geqslant\boldsymbol{0}$，则称$(\boldsymbol{M},\boldsymbol{N})$是一个**正则分裂**(regular splitting).若$\boldsymbol{N}\geqslant 0$换成$\boldsymbol{M}^{-1}\boldsymbol{N}\geqslant 0$，则称$(\boldsymbol{M},\boldsymbol{N})$为**弱正则分裂**.

如果$\boldsymbol{M}$非奇且$\boldsymbol{M}+\boldsymbol{N}$的对称部分是正定的，则称$(\boldsymbol{M},\boldsymbol{N})$是$\boldsymbol{A}$的一个$P$-**正则分裂**.这里$\boldsymbol{A}$的对称部分是指$\frac{1}{2}(\boldsymbol{A}+\boldsymbol{A}^{\mathrm{T}})$.

定义 9.11.3 设$\boldsymbol{A}=(a_{ij})\in\mathbb{R}^{n\times n}$的所有非对角元素$a_{ij}\leqslant 0$

$(i,j=1,2,\cdots,n,i\neq j)$，且 $\boldsymbol{A}$ 可逆，$\boldsymbol{A}^{-1}\geqslant 0$，则称 $\boldsymbol{A}$ 为 ***M*-矩阵**. 若记 $\langle \boldsymbol{A}\rangle=(\alpha_{ij})$，其中

$$\alpha_{ij}=\begin{cases}|a_{ij}|, & i\neq j,\\ -|a_{ij}|, & i\neq j,\end{cases}$$

如果 $\langle \boldsymbol{A}\rangle$ 为 M-矩阵，则称 A 为 ***H*-矩阵**.

由定义知，若 $\boldsymbol{A}$ 为 M-矩阵，则 $\boldsymbol{A}$ 一定是 H-矩阵，反之不成立.

定理 9.11.4 若下列三条件之一成立，则线性多分裂方法(9.11.4)收敛于方程(9.11.1)的解 $\boldsymbol{x}^*$.

(1) $\boldsymbol{A}^{-1}\geqslant 0$，且 $(\boldsymbol{M}_l,\boldsymbol{N}_l)(l=1,\cdots,L)$ 是 $\boldsymbol{A}$ 的弱正则分裂.

(2) $\boldsymbol{A}$ 对称正定，$(\boldsymbol{M}_l,\boldsymbol{N}_l)(l=1,2,\cdots,L)$ 是 $\boldsymbol{A}$ 的 P-正则分裂，且 $\boldsymbol{E}_l=a_l\boldsymbol{I}$（$a_l$ 为常数，$l=1,2,\cdots,L$）.

(3) $\|\boldsymbol{M}_l^{-1}\boldsymbol{N}_l\|_\infty<1(l=1,2,\cdots,L)$.

定理 9.11.5 设 $\boldsymbol{A}=(a_{ij})\in\mathbb{R}^{n\times n}$ 为 M-矩阵. $\boldsymbol{D}=\mathrm{diag}(a_{11},a_{22},\cdots,a_{nn})$，$\boldsymbol{A}=\boldsymbol{D}-\boldsymbol{C}$，且 $\boldsymbol{A}$ 的一个多分裂 $\{(\boldsymbol{M}_l,\boldsymbol{N}_l,\boldsymbol{E}_l)\mid l=1,2,\cdots,L\}$ 满足 $\boldsymbol{A}\leqslant\boldsymbol{M}_l\leqslant\boldsymbol{D}(l=1,2,\cdots,L)$，则有

$$\rho(\boldsymbol{G})=\rho\Big(\sum_{l=1}^{L}\boldsymbol{E}_l\boldsymbol{M}_l^{-1}\boldsymbol{N}_l\Big)\leqslant\rho(\boldsymbol{D}^{-1}\boldsymbol{C})<1. \qquad (9.11.7)$$

定理 9.11.5 表明在所设条件下，迭代法(9.11.4)收敛，注意式(9.11.7)对矩阵 $\boldsymbol{E}_l$ 的所有可能选取均成立.

在迭代法(9.11.4)的基础上，若引入松弛因子 $\omega>0$，可构造松弛多分裂方法

$$\boldsymbol{x}^{k+1}=\omega\sum_{l=1}^{L}\boldsymbol{E}_l\boldsymbol{M}_l^{-1}(\boldsymbol{N}_l\boldsymbol{x}^k+\boldsymbol{b})+(1-\omega)\boldsymbol{x}^k,\quad k=0,1,\cdots \qquad (9.11.8)$$

当 $\omega=1$ 时，式(9.11.8)即为式(9.11.4)，它也可写成

$$\boldsymbol{x}^{k+1}=\boldsymbol{G}_\omega\boldsymbol{x}^k+\boldsymbol{f}_\omega,$$

迭代矩阵 $\boldsymbol{G}_\omega$ 为

$$B = \langle a, r\rangle \cap D, \quad a \in D$$

的集合,如果存在非奇异矩阵 $\boldsymbol{\Lambda}=\boldsymbol{\Lambda}(B)$ 和正实数 $\lambda\in(0,1)$,有

$$\| \boldsymbol{x} - \boldsymbol{y} - \boldsymbol{\Lambda}(\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{y})) \| \leqslant \lambda \| \boldsymbol{\Lambda}(\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{y})) \|,$$

$\forall \boldsymbol{x},\boldsymbol{y}\in B$ 成立,则所有满足这些要求的函数 $\boldsymbol{f}$ 统称为函数类 F.若 $\boldsymbol{f}\in F$,可定义球形牛顿算子

$$N\boldsymbol{x} = \boldsymbol{x} - L\boldsymbol{f}(\boldsymbol{x}) = \langle \boldsymbol{x} - \boldsymbol{\Lambda f}(\boldsymbol{x}), \lambda \| \boldsymbol{\Lambda f}(\boldsymbol{x}) \| \rangle, \boldsymbol{x} \in B. \tag{9.10.5}$$

显然,$N\boldsymbol{x}$ 是一个球,且

$$\mathrm{mid}(N\boldsymbol{x}) = \boldsymbol{x} - \boldsymbol{\Lambda f}(\boldsymbol{x}), \quad \mathrm{rad}(N\boldsymbol{x}) = \lambda \| \boldsymbol{\Lambda f}(\boldsymbol{x}) \|.$$

对球形牛顿算子 N,也有类似于区间牛顿算子的性质,即

(1) 若 $\boldsymbol{x}^*\in B$,则 $\boldsymbol{f}(\boldsymbol{x}^*)=\boldsymbol{0}$ 的充分必要条件是 $\boldsymbol{x}^*=N\boldsymbol{x}^*$,求方程(9.10.1)的解等价于求 N 在 B 上的不动点;

(2) 若 $\boldsymbol{x}\in B$ 且 $B\cap N\boldsymbol{x}=\varnothing$,则方程(9.10.1)在 B 中无解;

(3) 若对 $\forall \boldsymbol{x}\in B$ 有 $N\boldsymbol{x}\in B$,则方程(9.10.1)在 B 中有解 $\boldsymbol{x}^*$ 存在;

(4) 若 $\boldsymbol{x}^*\in B$,$\boldsymbol{f}(\boldsymbol{x}^*)=0$,则 $\boldsymbol{x}^*\in N\boldsymbol{x}$.

利用球形牛顿算子 N 可构造**球形牛顿法**.

算法 9.10.3 (球形牛顿法)

初始步:取 $\boldsymbol{X}^0=D=\langle \boldsymbol{x}^0, r^0\rangle$ 计算 $N\boldsymbol{x}^0$.

1. 若 $\boldsymbol{X}^0\cap N\boldsymbol{x}^0=\varnothing$,则算法停止.

2. 若 $\boldsymbol{X}^0\cap N\boldsymbol{x}^0\neq\varnothing$,则定义交球(交球是指包含两球交的最小球)

$$\boldsymbol{X}^{(1)} = \langle \boldsymbol{X}^0 \cap N\boldsymbol{x}^0 \rangle.$$

计算步:假定 $\boldsymbol{X}^k(k\geqslant 1)$ 已定义,且 $\boldsymbol{x}=\mathrm{mid}\boldsymbol{X}^k\in\boldsymbol{X}^0$,已计算出了球 $N\boldsymbol{x}^k$.

1. 若 $\boldsymbol{X}^k\cap N\boldsymbol{x}^k=\varnothing$,则算法停止.

2. 若 $\boldsymbol{X}^k\cap N\boldsymbol{x}^k\neq\varnothing$,则新的球定义为

$$\bar{\boldsymbol{X}}^{k+1} = \langle \boldsymbol{X}^k \cap N\boldsymbol{X}^k \rangle$$

$$G_\omega = \omega \sum_{l=1}^{L} E_l M_l^{-1} N_l + (1-\omega) I$$

$$= \sum_{l=1}^{L} E_l M_l^{-1} [(1-\omega) M_l + \omega N_l]. \qquad (9.11.9)$$

定理 9.11.6 设 $A \in \mathbb{R}^{n\times n}$ 是一个 H-矩阵. $D = \text{diag}A$, 记 $J = |D|^{-1}C$, $\{(M_l, N_l, E_l) | l=1,2,\cdots,L\}$ 是 A 的一个多分裂. 且 $\text{diag}(M_l) = D$, $\langle A \rangle = \langle M_l \rangle - |N_l|$ $(l=1,\cdots,L)$, 则松弛多分裂方法(9.11.8)对任意初值 $x^0 \in \mathbb{R}^n$ 及 $\omega \in \left(0, \frac{2}{1+\rho(J)}\right)$ 都是收敛的.

由于 A 为 H-矩阵, 故 D 非奇异且 $\rho(J) < 1$, 可知 $1 < \frac{2}{1+\rho(J)} < 2$, 故 $\omega = 1$ 时迭代法收敛. 这就证明了多分裂迭代(9.11.4)收敛. 因此定理 9.11.6 包含了定理 9.11.5 的结论, 但条件减弱为 A 是 H-矩阵.

对不同的多分裂 $\{(M_l, N_l, E_l) | l=1,2,\cdots,L\}$ 可构造出各种不同的多分裂迭代法.

9.11.2 非线性多分裂方法

考虑非线性方程组

$$F(x) = 0, \qquad (9.11.10)$$

其中 $F: D \subset \mathbb{R}^n \to \mathbb{R}^n$, $F(x) = (f_1(x), \cdots, f_n(x))^T$.

定义 9.11.7 设 $F: D \subset \mathbb{R}^n \to \mathbb{R}^n$, 令 $F_l: D \times D \subset \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$, 使

$$F_l(x; x) = F(x) \quad (\forall x \in D \quad l = 1,2,\cdots,L), \qquad (9.11.11)$$

且 $E_l \in \mathbb{R}^{n\times n}$ $(l=1,2,\cdots,L)$ 为非负对角矩阵, 满足

$$\sum_{l=1}^{L} E_l = I,$$

则称二元组集合 $\{(F_l, E_l) | l=1,2,\cdots,L\}$ 为 $F(x)$ 的一个非线性多分裂, 对应的迭代法

$$\begin{cases} \boldsymbol{F}_l(\boldsymbol{x}^k;\boldsymbol{y}_l^k)=\boldsymbol{0}, \\ \boldsymbol{x}^{k+1}=\sum_{l=1}^{L}\boldsymbol{E}_l\boldsymbol{y}_l^k, \end{cases} (l=1,2,\cdots,L) \tag{9.11.12}$$

称为解方程(9.11.10)的**非线性多分裂方法**,简称 NM 方法.

定义 9.11.7 是定义 9.11.1 的推广,因为当 $\boldsymbol{F}(\boldsymbol{x})=\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}$ 时,若取 $\boldsymbol{F}_l(\boldsymbol{x};\boldsymbol{y})=\boldsymbol{M}_l\boldsymbol{y}-\boldsymbol{N}_l\boldsymbol{x}-\boldsymbol{b}(l=1,2,\cdots,L)$,迭代法(9.11.12)便是迭代法(9.11.4).

在形式上,与线性情形一样,NM 方法具有内在并行性,因为对每个不同的 l,$\boldsymbol{y}_l^k$ 的计算是彼此独立的,也适合于并行计算,而且 $\boldsymbol{E}_l$ 的第 i 个对角元素为零时 $\boldsymbol{y}_l^k$ 的第 i 个分量也无需计算.问题在于应用 NM 法时如何选取 $\boldsymbol{F}_1,\cdots,\boldsymbol{F}_l$ 对 $\boldsymbol{F}(x)$ 进行分裂及求方程 $\boldsymbol{F}_l(\boldsymbol{x}^k;\ \boldsymbol{y}^k)=\boldsymbol{0}$ 的解 $\boldsymbol{y}^k$.

对 $\boldsymbol{F}(\boldsymbol{x})$ 的非线性分裂有各种不同方法,这里只介绍其中一种较常用的.令整数集 $\mathrm{N}=\{1,2,\cdots,n\}$ 的非空子集 $S_l\subset\mathrm{N}$ $(l=1,2,\cdots,L)$,且 $\bigcup_{l=1}^{L}S_l=\mathrm{N}$,其中各 S_l 可以相交,对 $l=1,2,\cdots,L$ 定义 $\boldsymbol{E}_l=\mathrm{diag}(e_{l1},\cdots,e_{ln})$ 是非负对角矩阵,且 $e_{li}=0$.当 $i\notin S_l$ 时,对 $l=1,2,\cdots,L$ 定义映射 $\boldsymbol{P}_l$: $\mathbb{R}^n\to\mathbb{R}^n$,$\boldsymbol{P}_l=(P_{l1},\cdots,P_{ln})^{\mathrm{T}}$ 及 $\boldsymbol{F}_l$: $\mathbb{R}^n\times\mathbb{R}^n\to\mathbb{R}^n$,$\boldsymbol{F}_l(\boldsymbol{x};\boldsymbol{y})=(f_{l1}(\boldsymbol{x};\boldsymbol{y}),\cdots,f_{ln}(\boldsymbol{x};\boldsymbol{y}))^{\mathrm{T}}$,其中

$$P_{li}(\boldsymbol{x})=\begin{cases} x_i, & i\in S_l, \\ 0, & i\notin S_l, \end{cases}$$

$$f_{li}(\boldsymbol{x};\boldsymbol{y})=\begin{cases} f_i((\boldsymbol{I}-\boldsymbol{P}_l)\boldsymbol{x}+\boldsymbol{P}_l\boldsymbol{y}), & i\in S_l, \\ f_i(x_1,\cdots,x_{i-1},y_i,x_{i+1},\cdots,x_n), & i\notin S_l. \end{cases} \tag{9.11.13}$$

显然这样定义的 $\boldsymbol{F}_l(\boldsymbol{x};\boldsymbol{y})$ 满足 $\boldsymbol{F}_l(\boldsymbol{x};\boldsymbol{x})=\boldsymbol{F}(\boldsymbol{x})$,于是 $\{(\boldsymbol{F}_l,\boldsymbol{E}_l)\mid l=1,2,\cdots,L\}$ 是 $\boldsymbol{F}(\boldsymbol{x})$ 的一个多分裂,称为基于块 S_l 的非线性块多分裂,显然对 $i\notin S_l$ 的 f_{li} 的选取对相应 NM 法的结果无关紧要,具体说,计算 y_l^k 时注意的只是 $i\in S_l$ 的那些分量 y_{li}^k,因此,实际上需

要求解的是关于 $y_{li}^k(i\in S_l)$ 的子方程组 $f_{li}(\boldsymbol{x}^k;\boldsymbol{y}_l^k)=\boldsymbol{0}(i\in S_l)$. 式(9.11.13)中关于 $i\notin S_l$ 的 $f_{li}(\boldsymbol{x};\boldsymbol{y})$ 的选取只是使得 $\boldsymbol{F}_l(\boldsymbol{x};\boldsymbol{y})$ 与定义 9.11.7 一致的方案之一.

任何一种 NM 法(9.11.12)都要求 S_l 上的子方程组

$$\boldsymbol{F}_l(\boldsymbol{x}^k;\boldsymbol{y}_l^k)=\boldsymbol{0}\quad(l=1,2,\cdots,L)\qquad(9.11.14)$$

的解 $\boldsymbol{y}_l^k$,这些子方程仍需要用迭代法求其数值解,但子方程(9.11.14)只是 n_l 维的方程组,n_l 是子集 S_l 的元素个数,通常 $n_l\ll n$. 若使 $n_1=\cdots=n_L=m$,这样 L 个 m 维子方程可分别在 L 台处理机上并行求解,由于 m 较小,因此计算量及计算时间均可大大节省. 至于求解子方程(9.11.14)的方法,原则上可用前面已介绍的求解非线性方程组的任一种方法. 这里只给出牛顿法,非线性 SOR 法,SOR-Newton 法,Newton-SOR 等四种方法解(9.11.14)得到的 NM 法公式.

用 $\boldsymbol{F}'_{ly}(\boldsymbol{x};\boldsymbol{y})$ 表示 $\boldsymbol{F}_l(\boldsymbol{x};\boldsymbol{y})$ 在点$(\boldsymbol{x},\boldsymbol{y})$的雅可比矩阵

$$\boldsymbol{F}'_{ly}(\boldsymbol{x};\boldsymbol{y})=\begin{bmatrix}\dfrac{\partial f_{l1}(\boldsymbol{x};\boldsymbol{y})}{\partial y_1} & \cdots & \dfrac{\partial f_{l1}(\boldsymbol{x};\boldsymbol{y})}{\partial y_n}\\ \vdots & & \vdots\\ \dfrac{\partial f_{ln}(\boldsymbol{x};\boldsymbol{y})}{\partial y_1} & \cdots & \dfrac{\partial f_{ln}(\boldsymbol{x};\boldsymbol{y})}{\partial y_n}\end{bmatrix},\qquad(9.11.15)$$

并用 $\boldsymbol{D}_l^k,-\boldsymbol{L}_l^k,-\boldsymbol{U}_l^k$ 分别表示矩阵 $\boldsymbol{F}'_{ly}(\boldsymbol{x}^k;\ \boldsymbol{y}^k)$ 的对角,严格下三角,严格上三角部分,由于 $\boldsymbol{F}_l(\boldsymbol{x}^k;\boldsymbol{x}^k)=\boldsymbol{F}(\boldsymbol{x}^k)$,因此,如果从 $\boldsymbol{x}^k$ 出发用一步牛顿法求解方程组(9.11.14)产生的向量为 $\boldsymbol{y}_l^k$,则有

$$\begin{cases}\boldsymbol{y}_l^k=\boldsymbol{x}^k-[\boldsymbol{F}'_{ly}(\boldsymbol{x}^k;\boldsymbol{x}^k)]^{-1}\boldsymbol{F}(\boldsymbol{x}^k)\quad(l=1,\cdots,L),\\ \boldsymbol{x}^{k+1}=\displaystyle\sum_{l=1}^{L}E_l y_l^k\qquad\qquad(k=0,1,\cdots),\end{cases}\qquad(9.11.16)$$

称为**一步 NM-牛顿法**. 类似地,用一步的牛顿-SOR 迭代法(9.5.7)解方程组(9.11.14),则得

$$\begin{cases} \boldsymbol{y}_l^k = \boldsymbol{x}^k - \omega_l[\boldsymbol{D}_l^k - \omega_l \boldsymbol{L}_l^k]^{-1}\boldsymbol{F}(\boldsymbol{x}^k) & (l = 1,\cdots,L), \\ \boldsymbol{x}^{k+1} = \sum_{l=1}^{L} \boldsymbol{E}_l \boldsymbol{y}_l^k & (k = 0,1,\cdots), \end{cases}$$

(9.11.17)

称为一步 **NM-牛顿-SOR** 法. 若用非线性 SOR 迭代法求解方程组 (9.11.14), 则得

$$\begin{cases} \text{由 } f_{li}(x^k; y_{l1}^k, \cdots, y_{li-1}^k, y_{li}, x_{i+1}^k, \cdots, x_n^k) = 0, l = 1,\cdots,L, i \in S_l \\ \text{解出 } y_{li}, \text{令 } y_{li}^k = x_i^k + \omega_l(y_{li} - x_i^k), \quad \omega_l > 0 \\ \boldsymbol{x}^{k+1} = \sum_{l=1}^{L} \boldsymbol{E}_l \boldsymbol{y}_l^k, \quad k = 0,1,\cdots \end{cases}$$

(9.11.18)

称为 NM-SOR 法. 若用一步 SOR-牛顿法 (9.5.10) 解方程组 (9.11.14), 则得一步 NM-SOR-牛顿法:

$$\begin{cases} \text{对 } l = 1,2,\cdots,L \text{ 计算} \\ y_{li}^k = x_i^k - \omega_l \dfrac{f_{li}(x^k; y_{l1}^k, \cdots, y_{li-1}^k, x_i^k, \cdots, x_n^k)}{\dfrac{\partial}{\partial y_i} f_{li}(x^k; y_{l1}^k, \cdots, y_{li-1}^k, x_i^k, \cdots, x_n^k)}, \omega_l > 0, i \in S_l \\ \boldsymbol{x}^{k+1} = \sum_{l=1}^{L} \boldsymbol{E}_l \boldsymbol{y}_l^k, k = 0,1,\cdots \end{cases}$$

(9.11.19)

关于 NM 法的收敛性有以下的局部收敛定理.

定理 9.11.8 设 $\boldsymbol{F}$: $D \subset \mathbb{R}^n \to \mathbb{R}^n$ 的多分裂 $\{(\boldsymbol{F}_l, \boldsymbol{E}_l) \mid l = 1,\cdots,L\}$, $\boldsymbol{x}^*$ 为 $\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{0}$ 的零点, 假定映射 $\boldsymbol{F}_l(\boldsymbol{x}; \boldsymbol{y})$ $(l = 1,\cdots,L)$ 在点 $(\boldsymbol{x}^*, \boldsymbol{x}^*)$ 的邻域是连续可导的, 记 $\boldsymbol{M}_l = \boldsymbol{F}'_{ly}(\boldsymbol{x}; \boldsymbol{y})$, $\boldsymbol{N}_l = -\boldsymbol{F}'_{lx}(\boldsymbol{x}; \boldsymbol{y})$ 且 $\boldsymbol{M}_l$ 非奇异, 则 $\{(\boldsymbol{M}_l, \boldsymbol{N}_l, \boldsymbol{E}_l) \mid l = 1,\cdots,L\}$ 是 $\boldsymbol{F}'(\boldsymbol{x}^*)$ 的一个线性多分裂. 如果

$$\rho_1 = \rho\left(\sum_{l=1}^{L} \boldsymbol{E}_l \boldsymbol{M}_l^{-1} \boldsymbol{N}_l\right) < 1,$$

则 x^* 是 NM 法(9.11.12)和一步 NM-牛顿法(9.11.16)的一个吸引点,且在 x^* 的一个邻域中取初始近似 x^0. 上述两种迭代法均收敛到 x^*.

定理 9.11.9 在定理 9.11.8 的条件下,若令 $F_{l y}(x^*;x^*)=M_l=D_l-L_l-U_l$,其中 $D_l,-L_l,-U_l$ 分别记矩阵 M_l 的对角,严格下三角,严格上三角部分,且 $D_l(l=1,2,\cdots,L)$非奇异,记

$$B_l=\frac{1}{\omega_l}(D_l-\omega_l L_l),\quad C_l=\frac{1}{\omega_l}[(1-\omega_l)D_l+\omega_l U_l]$$

$$(\omega_l>0,l=1,\cdots,L),$$

则$\{(B_l,C_l+N_l,E_l)\,|\,l=1,\cdots,L\}$是 $F'(x^*)$的线性多分裂. 如果 $\rho_2=\rho\left(\sum_{l=1}^{L}E_l B_l^{-1}(C_l+N_l)\right)<1$,则 x^* 是迭代法(9.11.17),(9.11.18),(9.11.19)的吸引点,且只要初始近似在 x^* 一个足够好邻域中,则上述三个 NM 迭代法均收敛到 x^*.

9.12 无约束最优化方法

无约束最优化问题是一类很重要的实际应用问题,同时非线性方程组(9.1.2)的求解,也可转化为无约束最优化问题,本节只介绍有关的基本概念和常用算法.

9.12.1 基本概念

设 $f:D\subset\mathbb{R}^n\to\mathbb{R}^1$,在域 D 中求目标函数 f 的极小点问题,称为无约束最优化问题,记为

$$\min_{x\in D}f(x),\quad x=(x_1,\cdots,x_n)^{\mathrm{T}}\in\mathbb{R}^n.\qquad(9.12.1)$$

定义 9.12.1 若对 $x^*\in D\subset\mathbb{R}^n$,存在 $\delta>0$,使当 $\|x-x^*\|<\delta$ 时有

$$f(x)\geqslant f(x^*),\qquad(9.12.2)$$

则称 x^* 是 f 的**局部极小点**，若 $x \neq x^*$ 时式(9.12.2)为严格不等式，则称 x^* 为 f 的**严格局部极小点**.

定义 9.12.2 设 $x^* \in D \subset \mathbb{R}^n$，若对一切 $x \in D$，式(9.12.2)都成立，则称 x^* 是 f 的**全局极小点**，当 $x \neq x^*$ 时式(9.12.2)严格不等式成立，则称 x^* 为 f 的**严格全局极小点**.

例 9.12.3 设 $f: \mathbb{R}^2 \to \mathbb{R}^1$ 为

$$f(x) = 4 + 4.5x_1 - 4x_2 + x_1^2 + 2x_2^2 - 2x_1x_2 + x_1^4 - 2x_1^2x_2,$$

求 $\min f(x)$（通过作图得到）.

解 目标函数 f 的等高线如图 9.6 所示.

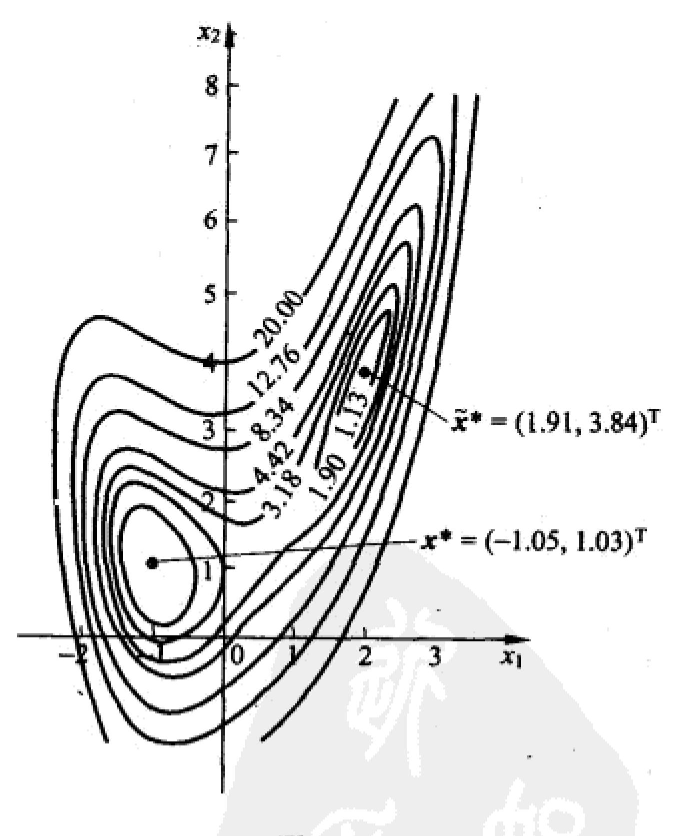


图 9.6

由于 $x^* = (-1.05, 1.03)^T$ 时 $f(x) \geqslant f(x^*) = 0.51$，$x$ 在 x^* 邻域，$\tilde{x}^* = (1.91, 3.84)^T$ 时 $f(x) \geqslant f(\tilde{x}^*) = 0.99$，$x$ 在 $\tilde{x}^*$ 邻域，

这表明 $\boldsymbol{x}^*$ 及 $\bar{\boldsymbol{x}}^*$ 均为局部极小点. 通过计算还知道在$\mathbb{R}^2$中 $f(\boldsymbol{x})\geqslant f(\boldsymbol{x}^*)$, 故 $\boldsymbol{x}^*$ 是$\mathbb{R}^2$中的全局极小点.

直接用定义检验 $\boldsymbol{x}^*$ 是否为局部极小点是很困难的,需要得到局部极小点的必要条件和充分条件才能建立计算方法. 根据多元函数的极值必要条件可得下面定理.

定理 9.12.4(必要条件)　假定 $f:D\subset\mathbb{R}^n\to\mathbb{R}^1$, $\boldsymbol{x}^*\in\mathrm{int}(D)$, f 在 $\boldsymbol{x}^*$ 的邻域内连续可导,且 $\boldsymbol{x}^*$ 为 f 的局部极小点,则

$$G(\boldsymbol{x}^*)=\nabla f(\boldsymbol{x}^*)=0, \tag{9.12.3}$$

其中 $\boldsymbol{G}(\boldsymbol{x})=\nabla f(\boldsymbol{x})=\left(\frac{\partial f(\boldsymbol{x})}{\partial x_1},\frac{\partial f(\boldsymbol{x})}{\partial x_2},\cdots,\frac{\partial f(\boldsymbol{x})}{\partial x_n}\right)^{\mathrm{T}}$ 称为 f 的梯度向量.

条件(9.12.3)是极小点的必要条件,满足条件(9.12.3)的点称为**稳定点**,它是否为极小点还要用到充分条件.

定理 9.12.5(充分条件)　设 $f:D\subset\mathbb{R}^n\to\mathbb{R}^1$二阶连续可导, $\boldsymbol{x}^*\in\mathrm{int}(D)$满足条件(9.12.3)且矩阵 $\boldsymbol{H}(\boldsymbol{x}^*)=\nabla^2 f(\boldsymbol{x}^*)$正定,则 $\boldsymbol{x}^*$ 是 f 的严格局部极小点.

这里 $\boldsymbol{H}$ 是 f 的黑塞(Hesse)矩阵.

$$\boldsymbol{H}(\boldsymbol{x})=\left(\frac{\partial^2 f(\boldsymbol{x})}{\partial x_i\partial x_j}\right)_{n\times n}$$

$\boldsymbol{H}(\boldsymbol{x}^*)$是对称的,如果对 $\forall\,\boldsymbol{y}\in\mathbb{R}^n$, $\boldsymbol{y}\neq\boldsymbol{0}$ 都有

$$(\boldsymbol{y},\boldsymbol{H}(\boldsymbol{x}^*)\boldsymbol{y})=\boldsymbol{y}^{\mathrm{T}}\boldsymbol{H}(\boldsymbol{x}^*)\boldsymbol{y}>0, \tag{9.12.4}$$

则称 $\boldsymbol{H}(\boldsymbol{x}^*)$**是正定的**.

如果 f 为正定二次函数 $f(\boldsymbol{x})=\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}^{\mathrm{T}}\boldsymbol{x}+c$, $\boldsymbol{A}\in\mathbb{R}^{n\times n}$对称正定, $\boldsymbol{x},\boldsymbol{b}\in\mathbb{R}^n$, c 为常数,则 $\min f(\boldsymbol{x})$有惟一极小点. 因为 $\boldsymbol{G}(\boldsymbol{x})=\nabla f(\boldsymbol{x})=\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}$, 由定理 9.12.4 知,极小点 $\boldsymbol{x}^*$ 满足 $\boldsymbol{A}\boldsymbol{x}^*-\boldsymbol{b}=\boldsymbol{0}$, 即 $\boldsymbol{x}^*=\boldsymbol{A}^{-1}\boldsymbol{b}$. 另一方面 $\boldsymbol{H}(\boldsymbol{x}^*)=\boldsymbol{A}$ 对称正定,故由定理 9.12.5 知 $\boldsymbol{x}^*$ 为 f 的惟一极小点.

9.12.2 下降算法与一维搜索

求无约束最优化问题(9.12.1)的一类重要迭代方法就是每一步迭代都使函数值 $f(\boldsymbol{x}^k)$减小,即

$$f(\boldsymbol{x}^{k+1}) < f(\boldsymbol{x}^k) \quad (k = 0,1,\cdots), \qquad (9.12.5)$$

这类算法称为**下降算法**. 其一般原则是从某一初始近似 $\boldsymbol{x}^0$ 出发,沿一个适当方向 $\boldsymbol{p}_0$,令 $\boldsymbol{x}^1 = \boldsymbol{x}^0 + \mu\boldsymbol{p}_0$,$\mu>0$,确定步长 $\mu=\mu_0$,使 $f(\boldsymbol{x}^1)=f(\boldsymbol{x}^0+\mu_0\boldsymbol{p}_0)<f(\boldsymbol{x}^0)$,一般情况是从 $\boldsymbol{x}^k$ 出发,沿 $f(\boldsymbol{x})$下降方向 $\boldsymbol{p}_k$ 用一维搜索方法选择步长 $\mu=\mu_k$,得到 $\boldsymbol{x}^{k+1}$,即

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \mu_k\boldsymbol{p}_k \quad (k = 0,1,\cdots), \qquad (9.12.6)$$

使式(9.12.5)成立,另一种选择 μ_k 的方法是使

$$f(\boldsymbol{x}^k + \mu_k\boldsymbol{p}_k) = \min_{\mu\geqslant 0} f(\boldsymbol{x}^k + \mu\boldsymbol{p}_k), \qquad (9.12.7)$$

这就是一维搜索的下降算法. 从算法步骤看到下降方向 $\boldsymbol{p}_k$ 和步长因子 μ_k 构成每一迭代步的修正量,是决定算法好坏的重要因素. 关于下降方向 $\boldsymbol{p}_k$ 的选择,以指向极小问题的最优解或使 f 下降最快的方向为佳. 以二维情形为例,若 f 的等高线族如图 9.7 所示,$\boldsymbol{x}^*$ 为极小点,图中给出三个方向 $\boldsymbol{l}_1,\boldsymbol{l}_2,\boldsymbol{l}_3$,显然,最理想的是取 $\boldsymbol{p}_k=\boldsymbol{l}_2$,因为它直指极小点方向. 但是,一般是难于求得的,通常可根据函数 f 的解析性质确定尽可能好的下降方向 $\boldsymbol{p}_k$.

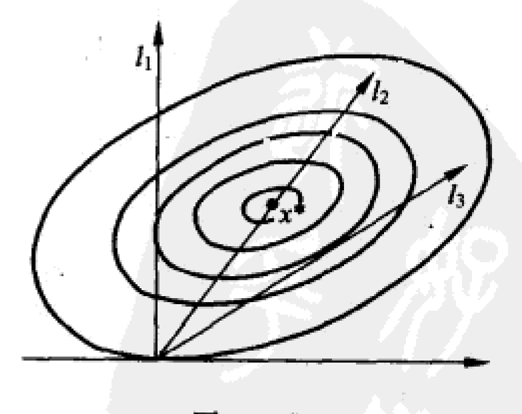


图 9.7

选择步长因子 μ_k 使式(9.12.7)成立就是一维搜索，它将多维目标函数转化为一维目标函数

$$\varphi(\mu)=f(\boldsymbol{x}^k+\mu\boldsymbol{p}_k) \tag{9.12.8}$$

的极小问题. 若 f 在域 D 中可导，即可由

$$\varphi'(\mu)=0 \tag{9.12.9}$$

确定稳定点，从而求得 μ_k. 解方程(9.12.9)的方法很多都可作为一维搜索算法，其中以二次插值法(即求根的抛物线法)收敛较快，是较常用的一维搜索算法. 设 $\varphi(\mu)$ 在 $\mu_i,\mu_{i-1},\mu_{i-2}$ 的函数值为 $\varphi_i,\varphi_{i-1},\varphi_{i-2}$，利用二次插值可求得

$$\mu_{i+1}=\frac{\mu_i+\mu_{i-1}}{2}-\frac{\varphi(\mu_i,\mu_{i-1})}{\varphi(\mu_i,\mu_{i-1},\mu_{i-2})},$$

并算出 $\varphi_{i+1}=\varphi(\mu_{i+1})$，再从 $\mu_i,\mu_{i-1},\mu_{i-2}$ 中去掉对应函数值最大的点代以新点 μ_{i+1}，得到三个新点，重复上述过程直到满足精度要求为止.

9.12.3 最速下降法与牛顿下降法

在下降算法中最古老和基本的数值方法是最速下降法，假定 $f(x)$ 二次连续可导，设 x^0 为初始近似，x^k 为 k 次近似，将 $f(x)$ 在 x^k 处接泰勒公式展开得

$$f(\boldsymbol{x}^k+\mu\boldsymbol{p}_k)=f(\boldsymbol{x}^k)+\mu\nabla f(\boldsymbol{x}^k)\boldsymbol{p}_k+o(\|\mu\boldsymbol{p}_k\|), \tag{9.12.10}$$

其中 $\nabla f(\boldsymbol{x}^k)$ 为函数 $f(\boldsymbol{x})$ 在 $\boldsymbol{x}^k$ 处的梯度向量，$o(\|\mu p_k\|)$ 对充分小的 μ 是高阶无穷小量，因此，要使不等式

$$f(\boldsymbol{x}^k+\mu p_k)<f(\boldsymbol{x}^k)$$

成立，式(9.11.10)中 μ 的一次项系数起决定作用，即忽略 μ 的高阶项，要求

$$\nabla f(\boldsymbol{x}^k)^{\mathrm{T}}p_k<0.$$

如果 $\nabla f(\boldsymbol{x}^k)\neq\boldsymbol{0}$，则使上式成立的 $\boldsymbol{p}_k$ 可以有无穷多个，但使

$|\nabla f(\boldsymbol{x}^k)^{\mathrm{T}}\boldsymbol{p}_k|$取最大值的 $\boldsymbol{p}_k$ 却只有一个,因为使

$$|\nabla f(\boldsymbol{x}^k)^{\mathrm{T}}\boldsymbol{p}_k| \leqslant \|\nabla f(\boldsymbol{x}^k)\| \|\boldsymbol{p}_k\|$$

成为等式,当且仅当 $\boldsymbol{p}_k=\nabla f(\boldsymbol{x}^k)$时才可能,此时

$$\nabla f(\boldsymbol{x}^k)^{\mathrm{T}}\nabla f(\boldsymbol{x}^k) = \|\nabla f(\boldsymbol{x}^k)\|^2,$$

亦即当 $\boldsymbol{p}_k=-\nabla f(\boldsymbol{x}^k)$时使$\nabla f(\boldsymbol{x}^k)\boldsymbol{p}_k$ 取到负最小值,这说明在 $\boldsymbol{x}^k$ 的适当小范围内,负梯度方向

$$\boldsymbol{p}_k = -\nabla f(\boldsymbol{x}^k)$$

是使 $f(\boldsymbol{x})$下降最快的方向. 此时,有

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \mu_k \nabla f(\boldsymbol{x}^k) \quad (k = 0,1,\cdots), \quad (9.12.11)$$

其中 μ_k 满足式(9.12.7)的要求,这样得到的算法就是**最速下降法**,也称**梯度法**. 这个方法的优点是程序简单,每步迭代计算量少,从一个不太好的初始近似 $\boldsymbol{x}^0$ 出发也能收敛到局部极小点,但是这个方法收敛慢,特别是 f 的黑塞矩阵 $\boldsymbol{H}(\boldsymbol{x})$呈病态情形时,收敛更慢,甚至花大量时间仍算不出要求的结果.

定理 9.12.6 设 $f:D\subset\mathbb{R}^n\to\mathbb{R}^1$ 在凸集 $D_0\subset D$ 上二次连续可微,其黑塞矩阵 $\boldsymbol{H}(\boldsymbol{x})$在 D_0 上满足

$$m\|\boldsymbol{y}\|^2 \leqslant \boldsymbol{y}^{\mathrm{T}}\boldsymbol{H}(\boldsymbol{x})\boldsymbol{y} \leqslant M\|\boldsymbol{y}\|^2,$$

其中 $M\geqslant m>0$,$\boldsymbol{y}\in\mathbb{R}^n$为任意非零向量,对任给的初始近似 $\boldsymbol{x}^0\in D_0$,集合 $S=\{\boldsymbol{x}\,|\,f(\boldsymbol{x})\leqslant f(\boldsymbol{x}^0)\}\subset D_0$ 有界,序列$\{\boldsymbol{x}^k\}$由最速下降法产生,则

(1) f 在 D_0 中存在惟一的全局极小点 $\boldsymbol{x}^*\in D_0$;

(2) $\{\boldsymbol{x}^k\}$收敛到 $\boldsymbol{x}^*$,且有估计式

$$\|\boldsymbol{x}^k-\boldsymbol{x}^*\| \leqslant c_2 q_0^{k/2}, \quad c_2<\infty,$$

$$f(\boldsymbol{x}^k)-f(\boldsymbol{x}^*) \leqslant q_0^k(f(\boldsymbol{x}^0)-f(\boldsymbol{x}^*)),$$

其中

$$q_0 = \left(1-\frac{m^2}{M^2}\right)<1.$$

最速下降法只有线性敛速,其收敛因子 q_0 依赖于矩阵 $\boldsymbol{H}(\boldsymbol{x})$

的条件数 $\rho=\frac{M}{m}$（M,m 分别表示 $\boldsymbol{H}(\boldsymbol{x})$ 的最大与最小特征值）. $q_0=1-\rho^{-2}$，要使 q_0 尽量小则要求 $\rho\approx1$；但是，若遇到 ρ 很大如 $\rho^{-1}\approx0$ 的情形，则 $q_0\approx1$，此时 $\boldsymbol{H}(\boldsymbol{x})$ 的最大与最小特征值相差很大，矩阵 $\boldsymbol{H}(\boldsymbol{x})$ 出现严重病态. 对 f 是二维正定二次函数而言，若其等高线族是一族很扁的椭圆，由最速下降法所得迭代序列 $\{\boldsymbol{x}^k\}$ 绕道前进，每次搜索方向严重偏离椭圆族中心 $\boldsymbol{x}^*$ 的方向，且每次迭代所跨的步长很小，如图 9.8，这时使用最速下降法效果极差.

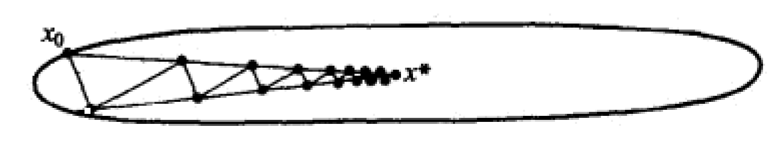


图 9.8

为了改善梯度法的收敛性，可将牛顿法用于解方程

$$\boldsymbol{G}(\boldsymbol{x})=\nabla f(\boldsymbol{x})=\boldsymbol{0}. \tag{9.12.12}$$

若 $\boldsymbol{H}(\boldsymbol{x})=\nabla^2 f(\boldsymbol{x})$ 正定，则可构造解最优化问题（9.12.1）的牛顿法

$$\boldsymbol{x}^{k+1}=\boldsymbol{x}^k-\boldsymbol{H}(\boldsymbol{x}^k)^{-1}\nabla f(\boldsymbol{x}^k) \quad (k=0,1,\cdots). \tag{9.12.13}$$

若记 $\boldsymbol{p}_k^N=-\boldsymbol{H}(\boldsymbol{x}^k)^{-1}\nabla f(\boldsymbol{x}^k)$，称为牛顿方向，而

$$\boldsymbol{H}(\boldsymbol{x}^k)\boldsymbol{p}_k^N=-\nabla f(\boldsymbol{x}^k) \tag{9.12.14}$$

称为牛顿方程. 由牛顿法的局部收敛定理 9.3.3 可知，序列 $\{\boldsymbol{x}^k\}$ 二阶收敛到 $\boldsymbol{x}^*$，特别是当 $f(x)$ 为二次函数，且 $H(x)$ 正定时，由于

$$\begin{aligned}\nabla f(\boldsymbol{x})&=\nabla f(\boldsymbol{x}^0)+\nabla^2 f(\boldsymbol{x}^0)(\boldsymbol{x}-\boldsymbol{x}^0)\\&=\nabla f(\boldsymbol{x}^0)+\boldsymbol{H}(\boldsymbol{x}^0)(\boldsymbol{x}-\boldsymbol{x}^0)=\boldsymbol{0}\end{aligned}$$

的解 $\boldsymbol{x}^*=\boldsymbol{x}^0-\boldsymbol{H}(\boldsymbol{x}^0)^{-1}\nabla f(\boldsymbol{x}^0)=\boldsymbol{x}^1$，说明用牛顿法求正定二次函数的极小点 $\boldsymbol{x}^*$，只需迭代一步就得到精确解 $\boldsymbol{x}^*$，具有这种有限迭代步求得正定二次函数极小的算法，称为具有二次终止性. 它表明

计算公式为

$$\begin{cases} \boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \mu_k \boldsymbol{p}_k, \\ \mu_k = -\dfrac{\nabla f(\boldsymbol{x}^k)^{\mathrm{T}} \boldsymbol{p}_k}{\boldsymbol{p}_k^{\mathrm{T}} \boldsymbol{H}(\boldsymbol{x}^k) \boldsymbol{p}_k} \text{ 或 } f(\boldsymbol{x}^k + \mu_k \boldsymbol{p}_k) = \min\limits_{\mu} f(\boldsymbol{x}^k + \mu \boldsymbol{p}_k), \\ \boldsymbol{p}_{k+1} = -\nabla f(\boldsymbol{x}^{k+1}) + \nu_k \boldsymbol{p}_k, \\ \nu_k = \dfrac{\nabla f(\boldsymbol{x}^{k+1})^{\mathrm{T}} \boldsymbol{H}(\boldsymbol{x}^k) \boldsymbol{p}_k}{\boldsymbol{p}_k^{\mathrm{T}} \boldsymbol{H}(\boldsymbol{x}^k) \boldsymbol{p}_k} \text{ 或 } \nu_k = \dfrac{\nabla f(\boldsymbol{x}^{k+1})^{\mathrm{T}} \nabla f(\boldsymbol{x}^{k+1})}{\nabla f(\boldsymbol{x}^k)^{\mathrm{T}} \nabla f(\boldsymbol{x}^k)}, \end{cases}$$

$$(k = 0, 1, \cdots) \tag{9.12.16}$$

由式(9.12.16)给出的共轭梯度法一般要计算 $\boldsymbol{H}(\boldsymbol{x}^k)$，如不方便可改用公式中不计算 $\boldsymbol{H}(\boldsymbol{x}^k)$ 的式子.

定理 9.12.7 若 $f(\boldsymbol{x})$ 满足定理 9.11.6 的全部条件，则由式(9.12.16)产生的序列 $\{\boldsymbol{x}^k\}$ 满足

(1) 序列 $\{f(\boldsymbol{x}^k)\}$ 为严格单调下降有下界序列；

(2) 序列 $\{\boldsymbol{x}^k\}$ 的极限点 $\boldsymbol{x}^* \in D$ 且 $\nabla f(\boldsymbol{x}^*) = \boldsymbol{0}$；

(3) $\boldsymbol{x}^* \in D$ 为最优化问题(9.12.1)的严格全局极小点.

9.12.5 变尺度法

为避免牛顿法及共轭梯度法计算二阶导数矩阵 $\boldsymbol{H}(\boldsymbol{x})$ 及求它的逆阵，可将求解非线性方程组的拟牛顿法用于解方程(9.12.12)，从而得到解最优化问题(9.12.1)的拟牛顿法，称为**变尺度法**，这类方法内容很多，以秩 2 拟牛顿法较为常用，下面只给出两个常用的算法. 令

$$\boldsymbol{s}_k = \boldsymbol{x}^{k+1} - \boldsymbol{x}^k, \boldsymbol{y}_k = \nabla f(\boldsymbol{x}^{k+1}) - \nabla f(\boldsymbol{x}^k), \boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \mu_k \boldsymbol{p}_k$$

$$(k = 0, 1, \cdots), \tag{9.12.17}$$

这里 $\boldsymbol{p}_k = -\boldsymbol{B}_k \nabla f(\boldsymbol{x}^k)$，$\mu_k$ 由一维搜索确定. 即

$$f(\boldsymbol{x}^k + \mu_k \boldsymbol{p}_k) = \min_{\mu} f(\boldsymbol{x}^k + \mu \boldsymbol{p}_k).$$

如果 $\boldsymbol{B}_k$ 的递推公式为

牛顿法具有二次终止性，当$\boldsymbol{H}(\boldsymbol{x}^k)$正定对称时由于$(\nabla f(\boldsymbol{x}^k),\boldsymbol{p}_k^N)=\nabla f(\boldsymbol{x}^k)^{\mathrm{T}}[-\boldsymbol{H}(\boldsymbol{x}^k)^{-1}\nabla f(\boldsymbol{x}^k)]<0$. 故选 $\boldsymbol{p}_k=\boldsymbol{p}_k^N$ 也是下降方向，于是可构造牛顿下降算法：

$$\boldsymbol{x}^{k+1}=\boldsymbol{x}^k-\mu_k\boldsymbol{H}(\boldsymbol{x}^k)^{-1}\nabla f(\boldsymbol{x}^k)\quad(k=0,1,\cdots),\tag{9.12.15}$$

μ_k 满足条件(9.12.7)，即

$$f(\boldsymbol{x}^k+\mu_k\boldsymbol{p}_k^N)=\min_{\mu>0}f(\boldsymbol{x}^k+\mu\boldsymbol{p}_k^N).$$

这种算法可用于改进梯度法的收敛性，但它每步要计算 f 的黑塞矩阵 $\boldsymbol{H}(\boldsymbol{x})$并求逆计算量较大，且当 $\boldsymbol{H}(\boldsymbol{x}^k)$的逆不存在或病态时 $\boldsymbol{p}_k$ 仍应选为负梯度方向 $\boldsymbol{p}_k=-\nabla f(\boldsymbol{x}^k)$.

9.12.4 共轭梯度法

共轭梯度法是求最优化问题最有效的方法之一，它较最速下降法更优越，对二次函数求极值不超过 n 步迭代即可求得精确解，故它是二次终止的算法. 方法是基于对函数 $f(x)$的二次逼近，当 f 满足定理 9.12.6 的条件时，取初始近似 $\boldsymbol{x}^0\in D_0,\boldsymbol{p}_0=-\nabla f(\boldsymbol{x}^0)$，将 $f(\boldsymbol{x}^0+\mu\boldsymbol{p}_0)$于 $\boldsymbol{x}_0$ 附近按泰勒公式展开，忽略三次项，得

$$f(\boldsymbol{x}^0+\mu\boldsymbol{p}_0)\approx f(\boldsymbol{x}^0)+\mu\nabla f(\boldsymbol{x}^0)^{\mathrm{T}}\boldsymbol{p}_0+\frac{1}{2}\mu^2\boldsymbol{p}_0^{\mathrm{T}}\boldsymbol{H}(\boldsymbol{x}^0)\boldsymbol{p}_0,$$

近似式右端为 μ 的二次函数，因 $\boldsymbol{H}(\boldsymbol{x}^0)$正定，即 $\boldsymbol{p}_0^{\mathrm{T}}\boldsymbol{H}(\boldsymbol{x}^0)\boldsymbol{p}_0>0$，故使该二次函数取极小的 μ 为

$$\mu_0=-\frac{\nabla f(\boldsymbol{x}^0)^{\mathrm{T}}\boldsymbol{p}_0}{\boldsymbol{p}_0^{\mathrm{T}}\boldsymbol{H}(\boldsymbol{x}^0)\boldsymbol{p}_0}.$$

若令 $\boldsymbol{x}^1=\boldsymbol{x}^*+\mu_0\boldsymbol{p}_0,\boldsymbol{p}_1=-\nabla f(\boldsymbol{x}^1)+\nu_0\boldsymbol{p}_0$ 要求满足 $\boldsymbol{p}_1^{\mathrm{T}}\boldsymbol{H}(\boldsymbol{x}^0)\boldsymbol{p}_0=0$，则得

$$\nu_0=\frac{\nabla f(\boldsymbol{x}^1)^{\mathrm{T}}\boldsymbol{H}(\boldsymbol{x}^0)\boldsymbol{p}_0}{\boldsymbol{p}_0^{\mathrm{T}}\boldsymbol{H}(\boldsymbol{x}^0)\boldsymbol{p}_0}.$$

若已求得 $\boldsymbol{x}^0,\boldsymbol{x}^1,\cdots,\boldsymbol{x}^k$ 及 $\boldsymbol{p}_0,\boldsymbol{p}_1,\cdots,\boldsymbol{p}_k$，则一般的共轭梯度法

$$\boldsymbol{B}_{k+1} = \boldsymbol{B}_k - \frac{\boldsymbol{B}_k \boldsymbol{y}_k \boldsymbol{y}_k^{\mathrm{T}} \boldsymbol{B}_k}{\boldsymbol{y}_k^{\mathrm{T}} \boldsymbol{B}_k \boldsymbol{y}_k} + \frac{\boldsymbol{s}_k \boldsymbol{s}_k^{\mathrm{T}}}{\boldsymbol{s}_k^{\mathrm{T}} \boldsymbol{y}_k} \quad (k = 0, 1, \cdots), \qquad (9.12.18)$$

则称此公式为 **DFP**(Davidon-Fletcher-Powell)**算法**.

若 $\boldsymbol{B}_k$ 的递推公式取为

$$\boldsymbol{B}_{k+1} = \left(\boldsymbol{I} - \frac{\boldsymbol{s}_k \boldsymbol{y}_k^{\mathrm{T}}}{\boldsymbol{s}_k^{\mathrm{T}} \boldsymbol{y}_k}\right) \boldsymbol{B}_k \left(\boldsymbol{I} - \frac{\boldsymbol{s}_k \boldsymbol{y}_k^{\mathrm{T}}}{\boldsymbol{s}_k^{\mathrm{T}} \boldsymbol{y}_k}\right)^{\mathrm{T}} + \frac{\boldsymbol{s}_k \boldsymbol{s}_k^{\mathrm{T}}}{\boldsymbol{s}_k^{\mathrm{T}} \boldsymbol{y}_k} \quad (k = 0, 1, \cdots), \qquad (9.12.19)$$

则称为 **BFGS**(Broyden-Fletcher-Goldfarb-Shanno)**算法**.

上述两种算法的搜索方向 $\boldsymbol{p}_k = -\boldsymbol{B}_k \nabla f(\boldsymbol{x}^k)$,当 $\boldsymbol{B}_k$ 为对称正定时 $\boldsymbol{B}_k^{-1} \boldsymbol{p}_k = -\nabla f(\boldsymbol{x}^k)$,表明是在 $\boldsymbol{B}_k^{-1}$ 度量意义下的最速下降方向. 在迭代过程中 $\boldsymbol{B}_k^{-1}$ 是不断变化的,故两种算法均为变尺度算法,方法的搜索方向是下降方向,且方法具有二次终止性. 计算时初始矩阵 $\boldsymbol{B}_0$ 可取为对称正定矩阵 $\boldsymbol{H}(\boldsymbol{x}^0)^{-1}$ 也可取 $\boldsymbol{B}_0 = \boldsymbol{I}$(单位矩阵).

9.13 非线性最小二乘法

在 9.1.2 节中已经指出用最小二乘法处理实验数据的曲线拟合问题时,关于参量的数学模型为非线性就是非线性最小二乘问题(9.1.8),解超定非线性方程组(9.1.10),也可转化为非线性最小二乘问题,设 $\boldsymbol{f}: D \subset \mathbb{R}^n \to \mathbb{R}^m$, $\boldsymbol{f} = (f_1, \cdots, f_m)^{\mathrm{T}}$,定义函数

$$\varphi(\boldsymbol{x}) = \frac{1}{2} \boldsymbol{f}(\boldsymbol{x})^{\mathrm{T}} \boldsymbol{f}(\boldsymbol{x}), \qquad (9.13.1)$$

显然 $\varphi: D \subset \mathbb{R}^n \to \mathbb{R}^1$,非线性最小二乘问题是求

$$\min_{\boldsymbol{x} \in D} \varphi(\boldsymbol{x}) = \min_{\boldsymbol{x} \in D} \frac{1}{2} \boldsymbol{f}(\boldsymbol{x})^{\mathrm{T}} \boldsymbol{f}(\boldsymbol{x}), \qquad (9.13.2)$$

即求目标函数 φ 的最优化问题. 因此它可用 9.12 节讨论的所有最优化方法求解. 但是由于问题(9.13.1)目标函数的特殊形式. 有可能得到便于计算的特殊算法. 根据极值必要条件,若 $\boldsymbol{f}$ 在 D 上可微,由式(9.13.1)对 φ 求导,得

$$\boldsymbol{G}(\boldsymbol{x}) = \nabla\varphi(\boldsymbol{x}) = \boldsymbol{D}\boldsymbol{f}(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}, \qquad (9.13.3)$$

称为**法方程**,其中

$$\boldsymbol{D}\boldsymbol{f}(\boldsymbol{x})^{\mathrm{T}} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_2}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_1} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_1}{\partial x_n} & \frac{\partial f_2}{\partial x_n} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix}.$$

为构造求解方程(9.13.3)的迭代法,可在点 $\boldsymbol{x}^k \in D$ 附近求 $\boldsymbol{f}(\boldsymbol{x})$的线性近似 $\boldsymbol{l}_k(\boldsymbol{x})$,若对 $\boldsymbol{f}(\boldsymbol{x})$做泰勒展开,其线性部分为

$$\boldsymbol{f}(\boldsymbol{x}) \approx \boldsymbol{D}\boldsymbol{f}(\boldsymbol{x}^k)(\boldsymbol{x} - \boldsymbol{x}^k) + \boldsymbol{f}(\boldsymbol{x}^k) = \boldsymbol{l}_k(\boldsymbol{x}).$$

若用 $\boldsymbol{l}_k(\boldsymbol{x})$代替式(9.12.3)中的 $\boldsymbol{f}(\boldsymbol{x})$,则得

$$\boldsymbol{D}\boldsymbol{l}_k(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{l}_k(\boldsymbol{x}) = \boldsymbol{0}, \qquad (9.13.4)$$

由此可得线性方程

$$\boldsymbol{D}\boldsymbol{f}(\boldsymbol{x}^k)^{\mathrm{T}}[\boldsymbol{D}\boldsymbol{f}(\boldsymbol{x}^k)(\boldsymbol{x} - \boldsymbol{x}^k) + \boldsymbol{f}(\boldsymbol{x}^k)] = \boldsymbol{0},$$

此方程解记作 $\boldsymbol{x}^{k+1}$,于是

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - [\boldsymbol{D}\boldsymbol{f}(\boldsymbol{x}^k)^{\mathrm{T}}\boldsymbol{D}\boldsymbol{f}(\boldsymbol{x}^k)]^{-1}\boldsymbol{D}\boldsymbol{f}(\boldsymbol{x}^k)^{\mathrm{T}}\boldsymbol{f}(\boldsymbol{x}^k) \quad (k = 0,1,\cdots)$$

称为高斯-牛顿法. 并可改写为

$$\begin{cases} \boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \boldsymbol{p}_k, \\ [\boldsymbol{D}\boldsymbol{f}(\boldsymbol{x}^k)^{\mathrm{T}}\boldsymbol{D}\boldsymbol{f}(\boldsymbol{x}^k)]\boldsymbol{p}_k = -\boldsymbol{D}\boldsymbol{f}(\boldsymbol{x}^k)^{\mathrm{T}}\boldsymbol{f}(\boldsymbol{x}^k), \end{cases} k = 0,1,\cdots \qquad (9.13.5)$$

注意此方法与直接用牛顿法解方程(9.13.3)不同,因为直接用牛顿法解要计算$\nabla\varphi$ 的导数,即 $\boldsymbol{f}$ 的二阶导数,计算较复杂,而式(9.13.5)只计算 $\boldsymbol{D}\boldsymbol{f}(\boldsymbol{x}^k)$,较为简单. 在式(9.13.5)中 $\boldsymbol{p}_k$ 是下降方向,它也可进行一维搜索. 只要引入搜索因子 μ_k,即

$$\begin{cases} \boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \mu_k\boldsymbol{p}_k, \\ \varphi(\boldsymbol{x}^k + \mu_k\boldsymbol{p}_k) = \min\limits_{\mu>0}\varphi(\boldsymbol{x}^k + \mu\boldsymbol{p}). \end{cases}$$

在使用时为了改善矩阵$[\boldsymbol{Df}(\boldsymbol{x}^k)^{\mathrm{T}}\boldsymbol{Df}(\boldsymbol{x}^k)]$的条件数，通常可引进一个参数$\mu_k\geqslant 0$，称为阻尼因子，将式(9.13.5)改为

$$\begin{cases}\boldsymbol{x}^{k+1}=\boldsymbol{x}^k+\boldsymbol{p}_k(\mu_k),\\ \boldsymbol{p}_k(\mu^k)=-[\boldsymbol{Df}(\boldsymbol{x}^k)^{\mathrm{T}}\boldsymbol{Df}(\boldsymbol{x}^k)+\mu_k\boldsymbol{I}]^{-1}\boldsymbol{Df}(\boldsymbol{x}^k)^{\mathrm{T}}\boldsymbol{f}(\boldsymbol{x}^k),\end{cases}\quad k=0,1,\cdots \tag{9.13.6}$$

称为阻尼最小二乘法.也称为**Levenberg-Marquardt** 法，它是高斯-牛顿法的改进.也是求解非线性最小二乘问题一个较常用的方法.算法中引进阻尼因子μ_k是使矩阵$[\boldsymbol{Df}(\boldsymbol{x}^k)^{\mathrm{T}}\boldsymbol{Df}(\boldsymbol{x}^k)+\mu_k\boldsymbol{I}]$的条件数得以改善并保证其正定性，$\mu_k>0$应使$\varphi(\boldsymbol{x}^{k+1})<\varphi(\boldsymbol{x}^k)$成立.可以证明当$\mu>0$，$\boldsymbol{p}_k(\mu)$是使$\varphi(\boldsymbol{x})$在$\boldsymbol{x}=\boldsymbol{x}^k$处下降的方向，因为此时

$$\boldsymbol{p}_k(\mu)^{\mathrm{T}}\nabla\varphi(\boldsymbol{x}^k)=-\boldsymbol{p}_k(\mu)^{\mathrm{T}}[\boldsymbol{Df}(\boldsymbol{x}^k)^{\mathrm{T}}\boldsymbol{Df}(\boldsymbol{x}_k)+\mu\boldsymbol{I}]\boldsymbol{p}_k(\mu)<0,$$

故只要选$\mu>0$，总可保证$\varphi(\boldsymbol{x}^{k+1})<\varphi(\boldsymbol{x}^k)$，但当$\mu>0$越大，序列$\{\boldsymbol{x}^k\}$收敛越慢，而若$\mu$太小，则收敛域过小，初始近似$\boldsymbol{x}^0$受限制.因此，适当选取$\mu_k$在计算中是很重要的，实际计算时可适当调节，具体方法见算法9.13.1.

算法 9.13.1 （阻尼最小二乘法）

(1) 置初始近似$\boldsymbol{x}^0$，误差限$\varepsilon>0$，阻尼因子μ_0（可取$\mu_0=10^{-2}$)，缩放常数ν(可取$\nu=2,5$或10).

(2) 计算$\boldsymbol{f}(\boldsymbol{x}^k)$，$\boldsymbol{Df}(\boldsymbol{x}^k)$，$\boldsymbol{Df}(\boldsymbol{x}^k)^{\mathrm{T}}\boldsymbol{Df}(\boldsymbol{x}^k)$，$\varphi(\boldsymbol{x}^k)$，$\nabla\varphi(\boldsymbol{x}^k)$.

(3) 解线性方程组求得$\boldsymbol{p}_k(\mu_k)=\boldsymbol{p}_k$，

$$[\boldsymbol{Df}(\boldsymbol{x}^k)^{\mathrm{T}}\boldsymbol{Df}(\boldsymbol{x}^k)+\mu_k\boldsymbol{I}]\boldsymbol{p}_k=-\boldsymbol{Df}(\boldsymbol{x}^k)^{\mathrm{T}}\boldsymbol{f}(\boldsymbol{x}^k).$$

(4) $\boldsymbol{x}^{k+1}=\boldsymbol{x}^k+\boldsymbol{p}_k(\mu_k)$，计算$\varphi(\boldsymbol{x}^{k+1})$.

(5) 若$\varphi(\boldsymbol{x}^{k+1})<\varphi(\boldsymbol{x}^k)$，取$\mu_k=\frac{\mu_k}{\nu}$转第(3)步求得新的$\bar{\boldsymbol{x}}^{k+1}$及$\varphi(\bar{\boldsymbol{x}}^{k+1})$，若$\varphi(\bar{\boldsymbol{x}}^{k+1})>\varphi(\boldsymbol{x}^k)$则转第(7)步，否则$\boldsymbol{x}^{k+1}=\bar{\boldsymbol{x}}^{k+1}$，$\varphi(\boldsymbol{x}^{k+1})=\varphi(\bar{\boldsymbol{x}}^{k+1})$转第(7)步.

(6) 若$\varphi(\boldsymbol{x}^{k+1})\geqslant\varphi(\boldsymbol{x}^k)$，取$\mu_k=\nu\mu_k$转第(3)步.

(7) 若$\|\boldsymbol{p}_k\|<\varepsilon$或$\|\nabla\varphi(\boldsymbol{x}^k)\|=\|\boldsymbol{Df}(\boldsymbol{x}^k)^{\mathrm{T}}\boldsymbol{f}(\boldsymbol{x}^k)\|\leqslant\varepsilon$.则终止，$\boldsymbol{x}^{k+1}\approx\boldsymbol{x}^*$，否则以$\boldsymbol{x}^{k+1}$代$\boldsymbol{x}^k$转第(2)步.

10 常微分方程初值问题的数值方法

10.1 引言

10.1.1 常微分方程的初值问题

一般 m 阶常微分方程的形式是

$$F(x,y,y',y'',\cdots,y^{(m)})=0. \tag{10.1.1}$$

如果函数 $y(x)$ 定义在区间 I 上，在 I 上具有 m 阶导数，且满足方程(10.1.1)，就称它是方程(10.1.1)的**解**.

如果方程(10.1.1)能写成

$$y^{(m)}=f(x,y,y',y'',\cdots,y^{(m-1)}), \tag{10.1.2}$$

则称方程写成**显式形式**. 如果方程(10.1.1)具有形式

$$a_m(x)y^{(m)}+a_{m-1}(x)y^{(m-1)}+\cdots+a_0(x)y=g(x), \tag{10.1.3}$$

则它是 ***m* 阶线性方程**(其中 $a_m(x)\neq 0$).

一般方程(10.1.1)有许多解，若要确定某个解就要外加一些条件. 本章主要讨论的一阶方程**初值问题**(initial value problem)是：

$$\frac{\mathrm{d}y}{\mathrm{d}x}=f(x,y),\quad x_0\leqslant x\leqslant b, \tag{10.1.4}$$

$$y(x_0)=y_0. \tag{10.1.5}$$

其中式(10.1.4)是微分方程，式中 f 是已知的函数. 式(10.1.5)是**初始条件**(initial condition)，式中的 y_0 是已知的值.

定理 10.1.1 设 f 为 x,y 的连续函数，则初值问题(10.1.4)，(10.1.5)在包含 x_0 的区间上有解 $y(x)$ 的充分必要条

件为

$$y(x) = y_0 + \int_{x_0}^{x} f(t, y(t))\mathrm{d}t. \qquad (10.1.6)$$

定义 10.1.2 如果存在常数 $L>0$,使函数 $f(x,y)$ 对所有区域 $D\subset\mathbb{R}^2$ 上的点 (x,y_1) 和 (x,y_2) 都有

$$| f(x,y_1) - f(x,y_2) | \leqslant L | y_1 - y_2 |, \qquad (10.1.7)$$

则称函数 $f(x, y)$ 在 D 上满足对变量 y 的**利普希茨条件**(Lipschitz condition),常数 L 称为 f 的**利普希茨常数**(Lipschitz constant).

假设函数 $f(x,y)$ 定义在凸域 $D\subset\mathbb{R}^2$ 上,且有对变量 y 的连续偏导数.如果存在常数 $L>0$,使得对所有 $(x,y)\in D$ 都有

$$\left|\frac{\partial f}{\partial y}(x,y)\right| \leqslant L,$$

则 f 在 D 上满足对变量 y 的利普希茨条件,其中的利普希茨常数就是 L.

定义 10.1.3 如果初值问题(10.1.4)、(10.1.5)满足:(1)存在惟一的连续可微的解.(2)对任意的 $\varepsilon>0$,存在正常数 $k(\varepsilon)$,使当 $|\varepsilon_0|<\varepsilon$ 且 $[x_0,b]$ 上连续函数 $\delta(x)$ 满足 $|\delta(x)|<\varepsilon$ 时,初值问题

$$\begin{cases}\dfrac{\mathrm{d}z}{\mathrm{d}x} = f(x,z) + \delta(x) \quad (x_0 \leqslant x \leqslant b),\\ z(x_0) = y_0 + \varepsilon_0\end{cases}$$

的解存在,且对一切 $x_0\leqslant x\leqslant b$ 满足

$$| z(x) - y(x) | < k(\varepsilon)\varepsilon,$$

则称初值问题(10.1.4),(10.1.5)是一个**适定的问题**(well-posed problem).

上述定义中(2)的意义是初值问题的解连续依赖于初值及方程右端的函数.

定理 10.1.4 设 f 在 $D=\{(x,y)\,|\,x_0\leqslant x\leqslant b, -\infty<y<+\infty\}$ 上连续,且满足对变量 y 的利普希茨条件,则初值问题

(10.1.4),(10.1.5)对任意的初值 y_0 是适定的.

一阶常微分方程组初值问题的一般形式为

$$\begin{cases}\frac{\mathrm{d}y_1}{\mathrm{d}x}=f_1(x,y_1,y_2,\cdots,y_m),y_1(x_0)=y_{10},\\ \frac{\mathrm{d}y_2}{\mathrm{d}x}=f_2(x,y_1,y_2,\cdots,y_m),y_2(x_0)=y_{20},\\ \vdots\\ \frac{\mathrm{d}y_m}{\mathrm{d}x}=f_m(x,y_1,y_2,\cdots,y_m),y_m(x_0)=y_{m0},\end{cases}\tag{10.1.8}$$

记向量 $\boldsymbol{y}=(y_1,y_2,\cdots,y_m)^{\mathrm{T}}$,$\boldsymbol{f}=(f_1,f_2,\cdots,f_m)^{\mathrm{T}}$,其中 $f_i=f_i(x,y_1,y_2,\cdots,y_m)(i=1,2,\cdots,m)$. $\boldsymbol{y}_0=(y_{10},y_{20},\cdots,y_{m0})^{\mathrm{T}}$,则方程组(10.1.8)的向量形式是

$$\begin{cases}\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}x}=\boldsymbol{f}(x,\boldsymbol{y})\quad(x_0\leqslant x\leqslant b),\\ \boldsymbol{y}(x_0)=\boldsymbol{y}_0.\end{cases}\tag{10.1.9}$$

$\boldsymbol{f}(x,\boldsymbol{y})$对变量 $\boldsymbol{y}$ 的利普希茨条件为:存在常数 $L\geqslant0$,使函数 $\boldsymbol{f}(x,\boldsymbol{y})$对 D 上的点$(x,\boldsymbol{y}_1)$和$(x,\boldsymbol{y}_2)$都有

$$\|\boldsymbol{f}(x,\boldsymbol{y}_1)-\boldsymbol{f}(x,\boldsymbol{y}_2)\|\leqslant L\|\boldsymbol{y}_1-\boldsymbol{y}_2\|,\tag{10.1.10}$$

其中的$\|\cdot\|$是$\mathbb{R}^m$中任一种范数.初值问题(10.1.9)的适定性概念和定理可以仿照定义 10.1.3 和定理 10.1.4 写出.

m 阶常微分方程初值问题的一般形式为

$$\begin{cases}\frac{\mathrm{d}^m y}{\mathrm{d}x^m}=f\left(x,y,\frac{\mathrm{d}y}{\mathrm{d}x},\cdots,\frac{\mathrm{d}^{m-1}y}{\mathrm{d}x^{m-1}}\right)\\ y(x_0)=y_0,\frac{\mathrm{d}y(x_0)}{\mathrm{d}x}=y_0^{(1)},\cdots,\frac{\mathrm{d}^{m-1}y(x_0)}{\mathrm{d}x^{m-1}}=y_0^{(m-1)}\end{cases}\tag{10.1.11}$$

如果设

$$y_1=y,y_2=\frac{\mathrm{d}y}{\mathrm{d}x},\cdots,y_m=\frac{\mathrm{d}^{m-1}y}{\mathrm{d}x^{m-1}},$$

则 m 阶方程的初值问题(10.1.11)可以化为方程组初值问题的

形式：

$$\begin{cases}\dfrac{dy_1}{dx}=y_2, & y_1(x_0)=y_0,\\ \quad\vdots & \quad\vdots \\ \dfrac{dy_{m-1}}{dx}=y_m, & y_{m-1}(x_0)=y_0^{(m-2)},\\ \dfrac{dy_m}{dx}=f(x,y_1,y_2,\cdots,y_m), & y_m(x_0)=y_0^{(m-1)}.\end{cases} \tag{10.1.12}$$

10.1.2 数值离散方法

考虑初值问题(10.1.4)，(10.1.5)在区间$[x_0,b]$上的离散解，引入点列$\{x_n\}$，满足

$$x_n=x_{n-1}+h_n \quad (n=1,2,\cdots),$$

其中$h_n>0$，称x_n为**节点**，h_n为**步长**. 通常考虑h_n为常数的情形，即$h_n=h(n=1,2,\cdots)$，这情形称为**等步长**的，此时有

$$x_n=x_0+nh \quad (n=1,2,\cdots). \tag{10.1.13}$$

取h使$\dfrac{b-x_0}{h}=N$为正整数，求$[x_0,b]$上的离散解，就是在点列$\{x_n\}_{n=0}^N$上求初值问题解的近似值. 以下记初值问题(10.1.4)，(10.1.5)的准确解$y(x)$在x_n点的值为$y(x_n)$，而在某一数值方法中$y(x_n)$的近似值记为y_n，并记$f_n=f(x_n,y_n)$.

定义 10.1.5 如果确定序列$\{y_n\}$的一个计算方法是由y_{n+j}，$f_{n+j}(j=0,1,\cdots,k)$的线性关系所组成. 即

$$\sum_{j=0}^k \alpha_j y_{n+j}=h\sum_{j=0}^k \beta_j f_{n+j}, \tag{10.1.14}$$

则称此方法为k步的**线性多步法**(linear multistep method)，其中，设$\alpha_k\neq 0$，且α_0,β_0两者不能全为0.

因为式(10.1.14)可以两边同乘一个常数因子，所以可规定

$\alpha_k=1$,这时线性多步法可写成

$$y_{n+k}=-\alpha_0 y_n-\alpha_1 y_{n+1}-\cdots-\alpha_{k-1}y_{n+k-1}+$$
$$h(\beta_0 f_n+\beta_1 f_{n+1}+\cdots+\beta_k f_{n+k}). \qquad (10.1.15)$$

如果已知 $y_n, y_{n+1}, \cdots, y_{n+k-1}$ 之值,就可通过式(10.1.15)计算 y_{n+k}. 在开始计算时,要用到计算的初值 $y_0, y_1, \cdots, y_{k-1}$,除了 y_0 是由初始条件(10.1.5)给出外,$y_1, \cdots, y_{k-1}$ 要通过别的办法给出. $y_0, y_1, \cdots, y_{k-1}$ 给出后,就可通过式(10.1.15)逐步计算 y_k, $y_{k+1}, \cdots$.

定义 10.1.6 从 y_n 的值计算 y_{n+1} 的方法,称为**单步法**(one-step method).

在方法(10.1.14)或(10.1.15)中,如果 $k=1$,就是线性的单步法. 一般的单步法可以写成

$$y_{n+1}=y_n+h\phi(x_n, y_n, y_{n+1}, h), \qquad (10.1.16)$$

在这样的单步法中,由初值问题的初始条件(10.1.5)可知 y_0 的值,然后即可逐步由式(10.1.16)计算 $y_1, y_2, \cdots$.

定义 10.1.7 在式(10.1.15)中,若 $\beta_k=0$,则称此计算方法是**显式的线性多步法**. 在式(10.1.16)中,若函数 ϕ 不显含 y_{n+1},则称此计算方法为**显式的单步法**. 这两者都属于**显式方法**(explicit method).

一般显式单步法可表示为

$$y_{n+1}=y_n+h\phi(x_n, y_n, h). \qquad (10.1.17)$$

定义 10.1.8 在式(10.1.15)中,若 $\beta_k\neq 0$,则称此计算方法是**隐式的线性多步法**. 在式(10.1.16)中,若函数中显含 y_{n+1},则称此计算方法是**隐式的单步法**. 这两者都属**隐式方法**(implicit method).

对于显式的单步法(10.1.17),可直接由 y_n 计算 y_{n+1}. 对于显式的线性多步法(10.1.15)(其中 $\beta_k=0$),也可直接由 y_{n+j}, f_{n+j}

$(j=0,1,\cdots,k-1)$直接计算 y_{n+k}.

对于隐式的单步法(10.1.16),如果已知 y_n,式(10.1.16)就是 y_{n+1} 的一个方程,可用解方程的方法求出 y_{n+1}. 对于隐式的线性多步法也同样,如果已知 $y_n, y_{n+1}, \cdots, y_{n+k-1}$,式(10.1.15)是 y_{n+k}的一个方程,可用解方程的方法求出 y_{n+1},例如牛顿法等. 也可以将式(10.1.15)改写为

$$y_{n+k} = h\beta_k f(x_{n+k}, y_{n+k}) + g, \tag{10.1.18}$$

其中 $g = \sum_{j=0}^{k-1}(-\alpha_j y_{n+j} + h\beta_j f_{n+j})$,它可以在计算 y_{n+k}之前预先计算好. 如果 f 是非线性函数,则式(10.1.18)是一个 y_{n+k} 的非线性方程,可以用下面的简单迭代法求解.

$$\begin{cases} \text{任给 } y_{n+k}^{[0]}, \\ y_{n+k}^{[s+1]} = h\beta_k f(x_{n+k}, y_{n+k}^{[s]}) + g, \quad s = 0,1,2,\cdots \end{cases} \tag{10.1.19}$$

迭代至 $|y_{n+k}^{[s+1]} - y_{n+k}^{[s]}| < \varepsilon$ 时结束,其中 ε 是预先给定的误差限.

定理 10.1.9 如果 $f(x,y)$满足对变量 y 的利普希茨条件,L 为其利普希茨常数,并设

$$h < \frac{1}{L\,|\beta_k|},$$

则由迭代法(10.1.19)得到的序列$\{y_{n+k}^{[s]}\}$收敛到方程(10.1.18)的解 y_{n+k}.

10.2 显式单步法的一般概念

初值问题(10.1.4),(10.1.5)的显式单步法是

$$y_{n+1} = y_n + h\phi(x_n, y_n, h). \tag{10.2.1}$$

定义 10.2.1 设 $y(x)$是初值问题(10.1.4),(10.1.5)的准确解,方法(10.2.1)是解此初值问题的一种显式单步法,则满足

$$y(x+h)-y(x)-h\phi(x,y(x),h)=O(h^{p+1}) \quad (10.2.2)$$

的最大整数 p 称为方法(10.2.1)的**阶**(order).

定义 10.2.2 如果

$$\phi(x,y,0)=f(x,y),$$

则称方法(10.2.1)与初值问题(10.1.4),(10.1.5)**相容**(consistency).

如果方法(10.2.1)与初值问题(10.1.4),(10.1.5)相容,利用泰勒公式在 $h=0$ 展开,可得

$$\begin{aligned}&y(x+h)-y(x)-h\phi(x,y(x),h)\\=&hy'(x)-h\phi(x,y(x),0)+O(h^2)\\=&O(h^2),\end{aligned}$$

所以,与初值问题相容的方法至少是一阶的方法.反之,p 阶($p\geqslant 1$)的方法(10.2.1)是相容的方法.

例 10.2.3 欧拉方法(Euler method).

$$y_{n+1}=y_n+hf(x_n,y_n) \quad (n=0,1,2,\cdots) \quad (10.2.3)$$

称欧拉方法.这里 $\phi(x,y,h)=f(x,y)$,所以欧拉方法与初值问题是相容的.又因为

$$\begin{aligned}&y(x+h)-y(x)-h\phi(x,y(x),h)\\=&y(x+h)-y(x)-hf(x,y(x))\\=&y(x+h)-y(x)-hy'(x)\\=&O(h^2),\end{aligned}$$

所以欧拉方法是一阶的显式单步法.

定义 10.2.4 初值问题(10.1.4),(10.1.5)的一种数值方法,若对任意的初值 y_0 及任意固定的 $x\in[x_0,b]$,$x=x_0+nh=x_n$,都有

$$\lim_{\substack{h\to 0\\(n\to\infty)}} y_n=y(x),$$

则称方法是**收敛**(convergence)的.

在实际计算中使用的方法，应该是相容的和收敛的.

定义 10.2.5 $e_{n+1}=y(x_{n+1})-y_{n+1}$ 称为数值方法在 x_{n+1} 点的**整体截断误差**(global truncation error).

定义 10.2.4 和定义 10.2.5 不单是对显式单步法的，对其他方法也适用. 对于显式单步法(10.2.1)有下面的定理.

定理 10.2.6 设 $h_0>0$，函数 $\phi(x,y,h)$ 在区域 $D=\{(x,y,h)|x\in[x_0,b],y\in(-\infty,+\infty),h\in[0,h_0]\}$ 上连续，且满足对变量 y 的利普希茨条件，则显式单步法(10.2.1)收敛的充分必要条件是它与初值问题相容；而且，若方法(10.2.1)是 p 阶的方法，则整体截断误差

$$y(x_n)-y_n=O(h^p).$$

定义 10.2.7

$$T_{n+1}=y(x_{n+1})-y(x_n)-h\phi(x_n,y(x_n),h) \qquad (10.2.4)$$

称为单步法(10.2.1)在 x_{n+1} 的**局部截断误差**(local truncation error)，其中 $y(x)$ 是初值问题(10.1.4)，(10.1.5)的准确解.

在计算 x_{n+1} 点近似解的值 y_{n+1} 时，如果假设 $y(x_n)=y_n$ 准确成立，即以上各步没有误差，则有 $T_{n+1}=y(x_{n+1})-y_{n+1}$，这就是“局部”的意义. 显然，$T_{n+1}$ 与 e_{n+1} 的意义是不相同的.

如果将式(10.2.4)中 T_{n+1} 的表达式按 h 的幂展开，则对于 p 阶的方法有

$$T_{n+1}=O(h^{p+1}).$$

定义 10.2.8 如果方法(10.2.1)是 p 阶方法，其局部截断误差可以写成

$$T_{n+1}=\psi(x_n,y(x_n))h^{p+1}+O(h^{p+2}),$$

则称 $\psi(x_n,y(x_n))h^{p+1}$ 为方法(10.2.1)在 x_{n+1} 的**主局部截断误差**(principal local truncation error).

常用的显式单步法，有欧拉方法、显式龙格-库塔(Runge-kutta)方法等.

10.3 欧拉方法

10.3.1 欧拉方法

例 10.2.3 已经给出初值问题(10.1.4),(10.1.5)的一种显式单步法——欧拉方法,即

$$y_{n+1} = y_n + hf(x_n, y_n) \quad (n = 0,1,2,\cdots). \quad (10.3.1)$$

欧拉方法可以看成在方程(10.1.4)中,$y'(x)$用差商$\dfrac{y(x+h)-y(x)}{h}$近似代替而得到的方法,它是一种最简单的显式方法,欧拉方法又可以看成在线性多步法的一般形式(10.1.15)中,令 $k=1$(即单步方法),系数取为 $\alpha_0=-1,\alpha_1=1,\beta_0=1,\beta_1=0$ 的情形,所以它是显式的线性单步法.

类似定理 10.1.1,把初值问题写成积分形式,有

$$y(x_{n+1}) = y(x_n) + \int_{x_n}^{x_{n+1}} f(t, y(t))\mathrm{d}t. \quad (10.3.2)$$

用 y_n 代替 $y(x_n)$,上式右端的积分若用左矩形数值积分公式近似计算,就可得到 $y(x_{n+1})$的近似值 y_{n+1},这就是公式(10.3.1),所以欧拉方法也可以看成是由数值积分公式构造出来的.

下面讨论欧拉方法的几何解释.设初值问题(10.1.4),(10.1.5)的准确解为 $y(x)$,其曲线过 $P_0(x_0,y_0)$点.欧拉方法可以看成从 P_0 出发,按照以 $f(x_0,y_0)$为斜率的方向作一直线段,向前推进到直线 $x=x_1$ 上一点 $P_1(x_1,y_1)$;再从 P_1 出发,按照以$f(x_1,y_1)$为斜率的方向作一直线段,推进到直线$x=x_2$上一点$P_2(x_2,y_2)$,依此类推.这样得到一条折线$\overline{P_0P_1P_2\cdots}$,作为曲线$y(x)$的近似曲线,如图 10.1.

欧拉方法是一阶的方法,它的局部截断误差为

$$T_{n+1} = \frac{h^2}{2}y''(x_n) + O(h^3).$$

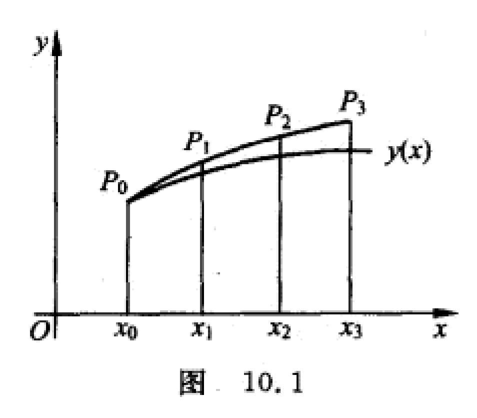


图 10.1

例 10.3.1 用欧拉方法近似求解初值问题

$$\begin{cases}\dfrac{\mathrm{d}y}{\mathrm{d}x}=-y+x+1 \quad (0\leqslant x\leqslant 1),\\ y(0)=1,\end{cases}$$

取步长 $h=0.1$.

解 用公式(10.3.1),其中 $x_n=nh=0.1n$,

$$\begin{aligned}&f(x_n,y_n)=-y_n+x_n+1,\\&y_0=1,\\&y_{n+1}=y_n+h(-y_n+x_n+1)\\&\qquad=(1-h)y_n+hx_n+h\\&\qquad=0.9y_n+0.1x_n+0.1 \quad (n=0,1,2,\cdots,9).\end{aligned}$$

计算结果及与准确解 $y(x)=x+\mathrm{e}^{-x}$ 的比较见表 10.1.

表 10.1

x_n	y_n	$y(x_n)$	$\lvert y(x_n)-y_n\rvert$
0	1.000000	1.000000	0
0.1	1.000000	1.004837	0.004837
0.2	1.010000	1.018731	0.008731
0.3	1.029000	1.040818	0.011818
0.4	1.056100	1.070320	0.014220
0.5	1.090490	1.106531	0.016041
0.6	1.131441	1.148812	0.017371
0.7	1.178297	1.196585	0.018288
0.8	1.230467	1.249329	0.018862
0.9	1.287420	1.306570	0.019150
1.0	1.348678	1.367879	0.019201

10.3.2 隐式欧拉方法和梯形方法

如果在公式(10.3.2)中，用 y_n 代替 $y(x_n)$，式中右端的积分式用右矩形数值积分公式近似计算，就得到如下的**隐式欧拉方法**(implicit Euler method)，或称**后退欧拉方法**(backward Euler method)：

$$y_{n+1} = y_n + hf(x_{n+1}, y_{n+1}). \qquad (10.3.3)$$

隐式欧拉方法(10.3.3)相当于在线性多步法一般形式(10.1.14)中，令 $k=1$，系数取为 $\alpha_0=-1, \alpha_1=1, \beta_0=0, \beta_1=1$，所以它是隐式的单步法. 如果设 $y_n=y(x_n)$，则有

$$y(x_{n+1}) - y_{n+1} = -\frac{h^2}{2}y''(x_n) + O(h^3).$$

它也是一阶的方法.

为了得到比欧拉方法更为精确的解，可以用数值积分方法中的梯形公式来近似式(10.3.2)右端的积分式，这样就得到如下解初值问题(10.1.4)，(10.1.5)的**梯形方法**(trapezoidal method)：

$$y_{n+1} = y_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1})]. \qquad (10.3.4)$$

梯形方法(10.3.4)相当于在线性多步法一般形式(10.1.14)中，令 $k=1$，系数取为 $\alpha_0=-1, \alpha_1=1, \beta_0=\beta_1=\frac{1}{2}$，所以梯形方法是一个隐式的单步法. 每步计算中为了求出 y_{n+1}，可以采用式(10.1.19)的迭代方法，其中的迭代初值 $y_{n+1}^{[0]}$ 就取为欧拉方法计算所得的值，即

$$\begin{cases} y_{n+1}^{[0]} = y_n + hf(x_n, y_n), \\ y_{n+1}^{[s+1]} = y_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1}^{[s]})] \quad (s = 0,1,2,\cdots) \end{cases} \qquad (10.3.5)$$

迭代到 $|y_{n+1}^{[s+1]} - y_{n+1}^{[s]}| < \epsilon$ 时结束，ϵ 为预先给定的误差限.

对梯形方法，如果设 $y_n=y(x_n)$，则有

$$y(x_{n+1})-y_{n+1}=-\frac{h^3}{12}y'''(x_n)+O(h^4).$$

梯形方法是二阶的方法.

10.3.3 改进的欧拉方法

在梯形方法求 y_{n+1} 的迭代(10.3.5)中，如果迭代只进行一次，$y_{n+1}^{[1]}$ 即作为 y_{n+1} 的值，这种计算方法称为**改进的欧拉方法**(modified Euler method)：

$$\begin{cases}\bar{y}_{n+1}=y_n+hf(x_n,y_n),\\ y_{n+1}=y_n+\dfrac{h}{2}[f(x_n,y_n)+f(x_{n+1},\bar{y}_{n+1})];\end{cases}\tag{10.3.6}$$

或者改写为

$$y_{n+1}=y_n+\frac{h}{2}[f(x_n,y_n)+f(x_{n+1},y_n+hf(x_n,y_n))].\tag{10.3.7}$$

改进的欧拉方法是一种属于式(10.2.1)形式的显式单步法，其中函数

$$\phi(x,y,h)=\frac{1}{2}[f(x,y)+f(x+h,y+hf(x,y))].$$

它的局部截断误差

$$T_{n+1}=O(h^3),$$

所以改进欧拉法是一种二阶的方法.

例 10.3.2 用改进的欧拉方法解初值问题

$$\begin{cases}\dfrac{\mathrm{d}y}{\mathrm{d}x}=-y+x+1\quad(0\leqslant x\leqslant 1),\\ y(0)=1,\end{cases}$$

取步长 $h=0.1$.

解 按公式(10.3.7)计算，对于本例可化成

$$y_{n+1} = 0.905y_n + 0.0095n + 0.1 \quad (n = 0,1,\cdots,9).$$

计算结果及与准确解 $y(x)=x+e^{-x}$ 的比较如表 10.2 所示.

表 10.2

x_n	y_n	$y(x_n)$	$\lvert y(x_n)-y_n \rvert$
0	1.000000	1.000000	0
0.1	1.005000	1.004837	1.63×10^{-4}
0.2	1.019025	1.018731	2.94×10^{-4}
0.3	1.041218	1.040818	4.00×10^{-4}
0.4	1.070802	1.070320	4.82×10^{-4}
0.5	1.107076	1.106531	5.45×10^{-4}
0.6	1.149404	1.148812	5.92×10^{-4}
0.7	1.197211	1.196585	6.26×10^{-4}
0.8	1.249976	1.249329	6.47×10^{-4}
0.9	1.307228	1.306570	6.58×10^{-4}
1.0	1.368541	1.367879	6.62×10^{-4}

10.4 龙格-库塔方法

10.4.1 显式龙格-库塔方法的一般形式

显式的龙格-库塔方法是一种显式单步法. R 级(R-stage)的显式龙格-库塔方法形式为

$$y_{n+1} = y_n + h\sum_{r=1}^{R} c_r K_r \quad (n = 0,1,\cdots), \tag{10.4.1}$$

其中

$$\begin{cases} K_1 = f(x_n, y_n), \\ K_r = f(x_n + a_r h, y_n + h\sum_{s=1}^{r-1} b_{rs} K_s) \quad (r = 2,3,\cdots,R) \end{cases} \tag{10.4.2}$$

10.4.3 三阶显式龙格-库塔方法

在式(10.4.1),式(10.4.2)中,设 $R=3,p=3$,同上述推导方法,得到

$$\begin{cases} c_1 + c_2 + c_3 = 1, \\ a_2 = b_{21}, \\ a_3 = b_{31} + b_{32}, \\ c_2 a_2 + c_3 a_3 = \dfrac{1}{2}, \\ c_2 a_2^2 + c_3 a_3^2 = \dfrac{1}{3}, \\ c_3 a_2 b_{32} = \dfrac{1}{6}. \end{cases} \tag{10.4.7}$$

这是 8 个未知数 6 个方程的方程组,可以得到多组不同的解.将满足方程组的 c_r, a_r, b_{rs} 代回到式(10.4.1)和式(10.4.2),就构成不同的显式三级三阶龙格-库塔方法.

例 10.4.4 休恩三阶方法(Heun's third order method).

在方程组(10.4.7)中,令 $c_1=\dfrac{1}{4}, c_2=0, c_3=\dfrac{3}{4}, a_2=b_{21}=\dfrac{1}{3}$, $b_{31}=0, a_3=b_{32}=\dfrac{2}{3}$,就得到

$$\begin{cases} y_{n+1} = y_n + \dfrac{h}{4}(K_1 + 3K_3), \\ K_1 = f(x_n, y_n), \\ K_2 = f\left(x_n + \dfrac{h}{3}, y_n + \dfrac{h}{3}K_1\right), \\ K_3 = f\left(x_n + \dfrac{2h}{3}, y_n + \dfrac{2h}{3}K_2\right). \end{cases} \tag{10.4.8}$$

例 10.4.5 库塔三阶方法(Kutta's third order method).

$$\begin{cases} c_1 + c_2 = 1 \\ a_2 c_2 = \dfrac{1}{2} \\ b_{21} c_2 = \dfrac{1}{2} \end{cases} \tag{10.4.3}$$

这是四个未知数(c_1, c_2, a_2, b_{21})三个方程的方程组,可以得到多组不同的解.将其解代回式(10.4.1)和式(10.4.2)中,就构成不同的二级二阶龙格-库塔方法.

例 10.4.1　中点方法(midpoint method).

在方程组(10.4.3)中令 $c_1 = 0, c_2 = 1, a_2 = b_{21} = \dfrac{1}{2}$,则式(10.4.1)和式(10.4.2)成为

$$y_{n+1} = y_n + hf\left(x_n + \frac{h}{2}, y_n + \frac{h}{2} f(x_n, y_n)\right). \tag{10.4.4}$$

式(10.4.4)可近似看成初值问题积分形式(10.3.2)中用中矩形数值积分公式近似积分式得到的结果.

例 10.4.2　改进的欧拉方法

在式(10.4.3)中,令 $c_1 = c_2 = \dfrac{1}{2}, a_2 = b_{21} = 1$,则方法(10.4.1),(10.4.2)成为

$$y_{n+1} = y_n + \frac{h}{2}[f(x_n, y_n) + f(x_n + h, y_n + hf(x_n, y_n))]. \tag{10.4.5}$$

这就是改进的欧拉方法(10.3.7).

例 10.4.3　休恩方法(Heun method).

在式(10.4.3)中,令 $c_1 = \dfrac{1}{4}, c_2 = \dfrac{3}{4}, a_2 = b_{21} = \dfrac{2}{3}$,则方法(10.4.1),(10.4.2)成为

$$y_{n+1} = y_n + \frac{h}{4}\left[f(x_n, y_n) + 3f\left(x_n + \frac{2}{3}h, y_n + \frac{2}{3}hf(x_n, y_n)\right)\right] \tag{10.4.6}$$

在方程组(10.4.7)中，令 $c_1=\frac{1}{6},c_2=\frac{2}{3},c_3=\frac{1}{6},a_2=b_{21}=\frac{1}{2},a_3=1,b_{31}=-1,b_{32}=2$，就得到

$$\begin{cases} y_{n+1}=y_n+\frac{h}{6}(K_1+4K_2+K_3), \\ K_1=f(x_n,y_n), \\ K_2=f\left(x_n+\frac{h}{2},y_n+\frac{h}{2}K_1\right), \\ K_3=f(x_n+h,y_n-hK_1+2hK_2). \end{cases} \tag{10.4.9}$$

10.4.4 四阶显式龙格-库塔方法

类似上述二阶、三阶方法的推导，可以构造多种四级四阶的龙格-库塔方法，其中最常见的是经典龙格-库塔方法.

例 10.4.6 经典龙格-库塔方法(classical Runge-Kutta method).

$$\begin{cases} y_{n+1}=y_n+\frac{h}{6}(K_1+2K_2+2K_3+K_4) \\ K_1=f(x_n,y_n) \\ K_2=f\left(x_n+\frac{1}{2}h,y_n+\frac{1}{2}hK_1\right) \\ K_3=f\left(x_n+\frac{1}{2}h,y_n+\frac{1}{2}hK_2\right) \\ K_4=f(x_n+h,y_n+hK_3) \end{cases} \tag{10.4.10}$$

例 10.4.7 用经典龙格-库塔方法解初值问题

$$\begin{cases} \frac{dy}{dx}=-y+x+1 \quad (0\leqslant x\leqslant 1), \\ y(0)=1. \end{cases}$$

取步长 $h=0.1$.

解 计算结果以及与准确解 $y(x)$ 的比较如表 10.3 所示.

表 10.3

x_n	y_n	$y(x_n)$	$\lvert y(x_n)-y_n\rvert$
0	1.00000000	1.00000000	0
0.1	1.00483750	1.00483742	8×10^{-8}
0.2	1.01873090	1.01873075	1.5×10^{-7}
0.3	1.04081842	1.04081822	2.0×10^{-7}
0.4	1.07032029	1.07032005	2.4×10^{-7}
0.5	1.10653093	1.10653066	2.7×10^{-7}
0.6	1.14881193	1.14881164	2.9×10^{-7}
0.7	1.19658562	1.19658530	3.2×10^{-7}
0.8	1.24932929	1.24932896	3.3×10^{-7}
0.9	1.30656999	1.30656966	3.3×10^{-7}
1.0	1.36787977	1.36787944	3.3×10^{-7}

四阶四级的龙格-库塔方法还有其他的例子.

例 10.4.8 基尔方法(Gill method).

$$\begin{cases} y_{n+1} = y_n + \dfrac{h}{6}[K_1 + (2-\sqrt{2})K_2 + (2+\sqrt{2})K_3 + K_4], \\ K_1 = f(x_n, y_n), \\ K_2 = f\left(x_n + \dfrac{h}{2}, y_n + \dfrac{h}{2}K_1\right), \\ K_3 = f\left(x_n + \dfrac{h}{2}, y_n + \dfrac{\sqrt{2}-1}{2}hK_1 + \left(1-\dfrac{\sqrt{2}}{2}\right)hK_2\right), \\ K_4 = f\left(x_n + h, y_n - \dfrac{\sqrt{2}}{2}hK_2 + \left(1+\dfrac{\sqrt{2}}{2}\right)hK_3\right). \end{cases} \tag{10.4.11}$$

如果希望得到 R 级的龙格-库塔方法是 R 阶的，则要确定系数 C_r，a_r 和 b_{rs}，使方法的局部截断误差 $T_{n+1}=O(h^{R+1})$. 把式(10.2.4)中的各项按 h 的幂展开，有

$$y(x_{n+1})=y(x_n)+hy'(x_n)+\frac{h^2}{2}y''(x_n)+\cdots+\frac{h^p}{p!}y^{(p)}(x_n)+O(h^{p+1}),$$

其中

$$y'(x_n)=f(x_n,y(x_n)),$$

$$y''(x_n)=\left[\frac{\partial f}{\partial x}+f\frac{\partial f}{\partial y}\right]_{(x_n,y(x_n))},$$

$$y'''(x_n)=\left[\frac{\partial^2 f}{\partial x^2}+2f\frac{\partial^2 f}{\partial x\partial y}+f^2\frac{\partial^2 f}{\partial y^2}+\frac{\partial f}{\partial y}\left(\frac{\partial f}{\partial x}+f\frac{\partial f}{\partial y}\right)\right]_{(x_n,y(x_n))},$$

$$\vdots$$

而 $y(x_n)+h\phi(x_n,y(x_n),h)$ 的展开，则需将式(10.4.1)，式(10.4.2)中函数 K_r，K_s 按二元函数的泰勒公式展开. 最后使 $T_{n+1}=O(h^{R+1})$，就可以构造出不同阶的龙格-库塔方法.

欧拉方法就是一阶一级的显式龙格-库塔方法.

10.4.2 二阶显式龙格-库塔方法

在式(10.4.1)和式(10.4.2)中，设 $R=2$，按照上述方法展开，有

$$y(x_{n+1})=y(x_n)+hf(x_n,y(x_n))+\frac{h^2}{2}\left[\frac{\partial f(x_n,y(x_n))}{\partial x}+f(x_n,y(x_n))\frac{\partial f(x_n,y(x_n))}{\partial y}\right]+O(h^3),$$

$$\begin{aligned}&y(x_n)+h[c_1K_1(x_n,y(x_n))+c_2K_2(x_n,y(x_n))]\\&=y(x_n)+h(c_1+c_2)f(x_n,y(x_n))+\\&h^2\left[a_2c_2\frac{\partial f(x_n,y(x_n))}{\partial x}+b_{21}c_2f(x_n,y(x_n))\frac{\partial f(x_n,y(x_n))}{\partial y}\right]+O(h^3),\end{aligned}$$

令上两式相减等于 $O(h^3)$，就有

例 10.4.9

$$\begin{cases} y_{n+1} = y_n + \frac{h}{8}(K_1 + 3K_2 + 3K_3 + K_4), \\ K_1 = f(x_n, y_n), \\ K_2 = f\left(x_n + \frac{1}{3}h, y_n + \frac{1}{3}hK_1\right), \\ K_3 = f\left(x_n + \frac{2}{3}h, y_n - \frac{1}{3}h(K_1 - 3K_2)\right), \\ K_4 = f(x_n + h, y_n + h(K_1 - K_2 + K_3)). \end{cases} \tag{10.4.12}$$

10.4.5 高阶显式龙格-库塔方法

R 级的显式龙格-库塔方法，在 $R=1,2,3,4$ 时，可以分别得到一、二、三、四阶的方法. 但是，通常 R 级的方法不一定是 R 阶的. 设 $p(R)$ 为 R 级显式方法可以达到的最大阶数，有如下结果：

当 $R=1,2,3,4$ 时，$p(R)=R$.

$p(5)=4, p(6)=5, p(7)=6, p(8)=6, p(9)=7$.

当 $R=10,11,\cdots, p(R)\leqslant R-2$.

例 10.4.10 库塔-尼斯特龙(Kutta-Nyström)五阶六级方法.

$$\begin{cases} y_{n+1} = y_n + \frac{h}{192}(23K_1 + 125K_3 - 81K_5 + 125K_6) \\ K_1 = f(x_n, y_n), \\ K_2 = f\left(x_n + \frac{1}{3}h, y_n + \frac{1}{3}hK_1\right), \\ K_3 = f\left(x_n + \frac{2}{5}h, y_n + \frac{1}{25}h(4K_1 + 6K_2)\right), \\ K_4 = f\left(x_n + h, y_n + \frac{1}{4}h(K_1 - 12K_2 + 15K_3)\right), \\ K_5 = f\left(x_n + \frac{2}{3}h, y_n + \frac{1}{81}h(6K_1 + 90K_2 - 50K_3 + 8K_4)\right), \\ K_6 = f\left(x_n + \frac{4}{5}h, y_n + \frac{1}{75}h(6K_1 + 36K_2 + 10K_3 + 8K_4)\right). \end{cases} \tag{10.4.13}$$

例 10.4.11 休塔(Huta)六阶八级方法.

$$
\begin{cases}
y_{n+1} = y_n + \dfrac{h}{840}(41K_1 + 216K_3 + 27K_4 + 272K_5 + \\
\qquad 27K_6 + 216K_7 + 41K_8), \\
K_1 = f(x_n, y_n), \\
K_2 = f\left(x_n + \dfrac{1}{9}h, y_n + \dfrac{1}{9}hK_1\right), \\
K_3 = f\left(x_n + \dfrac{1}{6}h, y_n + \dfrac{1}{24}h(K_1 + 3K_2)\right), \\
K_4 = f\left(x_n + \dfrac{1}{3}h, y_n + \dfrac{1}{6}h(K_1 - 3K_2 + 4K_3)\right), \\
K_5 = f\left(x_n + \dfrac{1}{2}h, y_n + \dfrac{1}{8}h(-5K_1 + 27K_2 - 24K_3 + 6K_4)\right), \\
K_6 = f\left(x_n + \dfrac{2}{3}h, y_n + \dfrac{1}{9}h(221K_1 - 981K_2 + \right. \\
\qquad \left. 867K_3 - 102K_4 + K_5)\right), \\
K_7 = f\left(x_n + \dfrac{5}{6}h, y_n + \dfrac{1}{48}h(-183K_1 + 678K_2 - \right. \\
\qquad \left. 472K_3 - 66K_4 + 80K_5 + 3K_6)\right), \\
K_8 = f\left(x_n + h, y_n + \dfrac{1}{82}h(716K_1 - 2079K_2 + 1002K_3 + \right. \\
\qquad \left. 834K_4 - 454K_5 - 9K_6 + 72K_7)\right).
\end{cases}
$$

10.4.6 龙格-库塔-费尔贝格方法

这是一种变步长控制误差的方法.因为在微分方程解比较平缓的区域,可以使用较大的步长,而变化较剧烈的区域,应该使用较小的步长,所以考虑**变步长**的方法,使各节点上整体截断误差 $|y(x_n)-y_n|(n=0,1,2,\cdots)$ 不超过某一允许值,而使用的节点尽量少.

由定理 10.2.6,一种 p 阶的方法,其局部截断误差 $T_{n+1}=O(h^{p+1})$,则其整体截断误差 $e_{n+1}=O(h^p)$.所以,如果 $\left|\dfrac{T_{n+1}}{h}\right|<\varepsilon$,

可有$|e_{n+1}|<K\epsilon$.

设有一种 p 阶的方法,从 x_n 到 x_{n+1} 采用步长 h_n,方法写成

$$\tilde{y}_{n+1} = \tilde{y}_n + h_n\tilde{\varphi}(x_n, \tilde{y}_n, h_n), \tag{10.4.14}$$

其局部截断误差 $\tilde{T}_{n+1}=O(h_n^{p+1})$. 再选择另一种$p+1$阶的方法,也用步长 h_n,方法写成

$$\hat{y}_{n+1} = \hat{y}_n + h_n\hat{\varphi}(x_n, \hat{y}_n, h_n), \tag{10.4.15}$$

其局部截断误差 $\hat{T}_{n+1}=O(h_n^{p+2})$. 记

$$d_{n+1} = \hat{y}_{n+1} - \tilde{y}_{n+1},$$

可有

$$\frac{d_{n+1}}{h} = \frac{\hat{y}_{n+1} - \tilde{y}_{n+1}}{h} \approx \frac{\tilde{T}_{n+1} - \hat{T}_{n+1}}{h} \approx ch^p.$$

在第 n 步计算中,先定一个 h 值,取 $h_n=h$,分别用方法(10.4.14)和方法(10.4.15)计算$\tilde{y}_{n+1}$和$\hat{y}_{n+1}$. 然后用 qh 代替 h 计算. 因为

$$\frac{d_{n+1}(qh)}{qh} \approx c(qh)^p \approx q^p\,\frac{\hat{y}_{n+1} - \tilde{y}_{n+1}}{h},$$

所以取 q 满足

$$|q| < \left(\frac{\epsilon h}{d_{n+1}}\right)^{1/p} \tag{10.4.16}$$

就可保证

$$\left|\frac{T_{n+1}(qh)}{qh}\right| < \epsilon.$$

新的步长就选为 qh,重新计算 y_{n+1}的值. 考虑到一个步长选择不当而要重新计算时有较大浪费,而上面的近似估计分析也要补偿,所以有时取

$$q = 0.8 \times \left|\frac{\epsilon h}{d_{n+1}}\right|^{1/p}.$$

实际计算时还有一些较详细的技术处理.

例 10.4.12 龙格-库塔-费尔贝格(Runge-Kutta-Fehlberg)**四阶方法**方法(10.4.14)和(10.4.15)分别取为以下的四阶及五阶公

式，其中用到的 $K_i(i=1,2,3,4,5)$ 都是相同的.

$$\tilde{y}_{n+1} = y_n + h\left(\frac{25}{216}K_1 + \frac{1408}{2565}K_3 + \frac{2197}{4104}K_4 - \frac{1}{5}K_5\right), \qquad (10.4.17)$$

$$\hat{y}_{n+1} = y_n + h\left(\frac{16}{135}K_1 + \frac{6656}{12825}K_3 + \frac{28561}{56430}K_4 - \frac{9}{50}K_5 + \frac{2}{55}K_6\right), \qquad (10.4.18)$$

其中

$$K_1 = f(x_n, y_n),$$

$$K_2 = f\left(x_n + \frac{h}{4}, y_n + \frac{h}{4}K_1\right),$$

$$K_3 = f\left(x_n + \frac{3h}{8}, y_n + \frac{3h}{32}K_1 + \frac{9h}{32}K_2\right),$$

$$K_4 = f\left(x_n + \frac{12h}{13}, y_n + \frac{1932h}{2197}K_1 - \frac{7200h}{2197}K_2 + \frac{7296h}{2197}K_3\right),$$

$$K_5 = f\left(x_n + h, y_n + \frac{439h}{216}K_1 - 8hK_2 + \frac{3680h}{513}K_3 - \frac{845h}{4104}K_4\right),$$

$$K_6 = f\left(x_n + \frac{h}{2}, y_n - \frac{8h}{27}K_1 + 2hK_2 - \frac{3544h}{2565}K_3 + \frac{1859h}{4104}K_4 - \frac{11h}{40}K_5\right).$$

一般选 q 为

$$q = \left(\frac{\varepsilon h}{2\,|\hat{y}_{n+1} - \tilde{y}_{n+1}|}\right)^{1/4} = 0.84 \times \left(\frac{\varepsilon h}{|\hat{y}_{n+1} - \tilde{y}_{n+1}|}\right)^{1/4}. \qquad (10.4.19)$$

具体计算时，先选定一个 h(或取为上一步的 h)，计算

$$\frac{|\hat{y}_{n+1} - \tilde{y}_{n+1}|}{h} = \left|\frac{1}{360}K_1 - \frac{128}{4275}K_3 - \frac{2197}{75240}K_4 + \frac{1}{50}K_5 + \frac{2}{55}K_6\right|.$$

如果 $\frac{|\hat{y}_{n+1} - \tilde{y}_{n+1}|}{h} < \varepsilon$，则按式(10.4.17)计算 y_{n+1}，转入下一步. 否则，按式(10.4.19)计算 q，再以 qh 为新的步长重新计算.

例 10.4.13　龙格-库塔-费尔贝格五阶方法. 方法(10.4.14)和(10.4.15)分别取为

$$\tilde{y}_{n+1} = y_n + h\sum_{k=1}^{6}\tilde{c}_k f_k,$$

$$\hat{y}_{n+1} = y_n + h\sum_{k=1}^{8}\hat{c}_k f_k,$$

其中

$$f_k = f\left(x_n + \alpha_k h, y_n + h\sum_{\lambda=1}^{k-1}\beta_{k\lambda} f_\lambda\right), \qquad (10.4.20)$$

式中系数如表 10.4,和式 $\sum_{\lambda=1}^{0}$ 认为是 0.

表 10.4

k \ λ	$\beta_{k\lambda}$							α_k	$\tilde{c}_k$	$\hat{c}_k$
	1	2	3	4	5	6	7			
1	0							0	$\frac{31}{380}$	$\frac{7}{1408}$
2	$\frac{1}{6}$							$\frac{1}{6}$	0	0
3	$\frac{4}{75}$	$\frac{16}{75}$						$\frac{4}{15}$	$\frac{1125}{2816}$	$\frac{1125}{2816}$
4	$\frac{5}{6}$	$-\frac{8}{3}$	$\frac{5}{2}$					$\frac{2}{3}$	$\frac{9}{32}$	$\frac{9}{32}$
5	$-\frac{8}{5}$	$\frac{144}{25}$	-4	$\frac{16}{25}$				$\frac{4}{5}$	$\frac{125}{768}$	$\frac{125}{768}$
6	$\frac{361}{320}$	$-\frac{18}{5}$	$\frac{407}{128}$	$-\frac{11}{80}$	$\frac{55}{128}$			1	$\frac{5}{66}$	0
7	$-\frac{11}{640}$	0	$\frac{11}{256}$	$-\frac{11}{160}$	$\frac{11}{256}$	0		0	—	$\frac{5}{66}$
8	$\frac{93}{640}$	$-\frac{18}{5}$	$\frac{803}{256}$	$-\frac{11}{160}$	$\frac{99}{256}$	0	1	1	—	$\frac{5}{66}$

例 10.4.14　龙格-库塔-费尔贝格六阶方法. 方法(10.4.14)和(10.4.15)分别取为

$$\tilde{y}_{n+1} = y_n + h\sum_{k=1}^{8}\tilde{c}_k f_k,$$

$$\hat{y}_{n+1} = y_n + h\sum_{k=1}^{10}\hat{c}_k f_k,$$

其中 f_k 的表达式同例 10.4.12 的式(10.4.20),系数如表 10.5 所示.

表 10.5

λ / k	$\beta_{k\lambda}$ 1	2	3	4	5	6	7	8	9	a_k	$\tilde{c}_k$	$\hat{c}_k$
1	0									0	$\frac{77}{1440}$	$\frac{11}{864}$
2	$\frac{2}{33}$									$\frac{2}{33}$	0	0
3	0	$\frac{4}{33}$								$\frac{4}{33}$	0	0
4	$\frac{1}{22}$	0	$\frac{3}{22}$							$\frac{2}{11}$	$\frac{1771561}{6289920}$	$\frac{1771561}{6289920}$
5	$\frac{43}{64}$	0	$-\frac{165}{64}$	$\frac{77}{32}$						$\frac{1}{2}$	$\frac{32}{105}$	$\frac{32}{105}$
6	$-\frac{2383}{486}$	0	$\frac{1067}{54}$	$-\frac{26312}{1701}$	$\frac{2176}{1701}$					$\frac{2}{3}$	$\frac{243}{2560}$	$\frac{243}{2560}$
7	$\frac{10077}{4802}$	0	$-\frac{5643}{686}$	$\frac{116259}{16807}$	$-\frac{6240}{16807}$	$\frac{1053}{2401}$				$\frac{6}{7}$	$\frac{16807}{74880}$	$\frac{16807}{74880}$
8	$-\frac{733}{176}$	0	$\frac{141}{8}$	$-\frac{335763}{23296}$	$\frac{216}{77}$	$-\frac{4617}{2816}$	$\frac{7203}{9152}$			1	$\frac{11}{270}$	0
9	$\frac{15}{352}$	0	0	$-\frac{5445}{46592}$	$\frac{18}{77}$	$-\frac{1215}{5632}$	$\frac{1029}{18304}$	0		0	—	$\frac{11}{270}$
10	$-\frac{1833}{352}$	0	$\frac{141}{8}$	$-\frac{51237}{3584}$	$\frac{18}{7}$	$-\frac{729}{512}$	$\frac{1029}{1408}$	0	1	1	—	$\frac{11}{270}$

龙格-库塔-费尔贝格七阶方法

方法(10.4.14)和(10.4.15)分别取为

$$\tilde{y}_{n+1} = y_n + \sum_{k=1}^{11} \tilde{c}_k f_k,$$

$$\hat{y}_{n+1} = y_n + \sum_{k=1}^{13} \hat{c}_k f_k,$$

其中 f_k 的表达式同例 10.4.12 的式(10.4.20),系数如表 10.6 所示.

表 10.6

k \ λ	$\beta_{k\lambda}$ 1	2	3	4	5	6	7	8	9	10	11	12	a_k	$\bar{c}_k$	$\hat{c}_k$
1	0												0	$\frac{41}{840}$	0
2	$\frac{2}{27}$												$\frac{2}{27}$	0	0
3	$\frac{1}{36}$	$\frac{1}{12}$											$\frac{1}{9}$	0	0
4	$\frac{1}{24}$	0	$\frac{1}{8}$										$\frac{1}{6}$	0	0
5	$\frac{5}{12}$	0	$-\frac{25}{16}$										$\frac{5}{12}$	0	0
6	$\frac{1}{20}$	0	0	$\frac{1}{4}$	$\frac{1}{5}$								$\frac{1}{2}$	$\frac{34}{105}$	$\frac{34}{105}$
7	$-\frac{25}{108}$	0	0	$\frac{125}{108}$	$-\frac{65}{27}$	$\frac{125}{54}$							$\frac{5}{6}$	$\frac{9}{35}$	$\frac{9}{35}$
8	$\frac{31}{300}$	0	0	0	$\frac{61}{225}$	$-\frac{2}{9}$	$\frac{13}{900}$						$\frac{1}{6}$	$\frac{9}{35}$	$\frac{9}{35}$
9	2	0	0	$-\frac{53}{6}$	$\frac{704}{45}$	$-\frac{107}{9}$	$\frac{67}{90}$	3					$\frac{2}{3}$	$\frac{9}{280}$	$\frac{9}{280}$
10	$-\frac{91}{108}$	0	0	$\frac{23}{108}$	$-\frac{976}{135}$	$\frac{311}{54}$	$-\frac{19}{60}$	$\frac{17}{6}$	$-\frac{1}{12}$				$\frac{1}{3}$	$\frac{9}{280}$	$\frac{9}{280}$
11	$\frac{2383}{4100}$	0	0	$-\frac{341}{164}$	$\frac{4496}{1025}$	$-\frac{301}{82}$	$\frac{2133}{4100}$	$\frac{45}{82}$	$\frac{45}{164}$	$\frac{18}{41}$			1	$\frac{41}{840}$	0
12	$\frac{3}{205}$	0	0	0	0	$-\frac{6}{41}$	$-\frac{3}{205}$	$-\frac{3}{41}$	$\frac{3}{41}$	$\frac{6}{41}$	0		0	—	$\frac{41}{840}$
13	$-\frac{1777}{4100}$	0	0	$-\frac{341}{164}$	$\frac{4496}{1025}$	$-\frac{289}{82}$	$\frac{2193}{4100}$	$\frac{51}{82}$	$\frac{33}{164}$	$\frac{12}{41}$	0	1	1	—	$\frac{41}{840}$

10.4.7 隐式龙格-库塔方法

R 级**隐式龙格-库塔方法**(implicit Runge-Kutta methods)的一般形式为

$$\begin{cases} y_{n+1} = y_n + h\sum_{r=1}^{R} c_r K_r, & (10.4.21) \\ K_r = f\left(x_n + a_r h, y_n + h\sum_{s=1}^{R} b_{rs} K_s\right) \\ \qquad (r = 1,2,\cdots,R). & (10.4.22) \end{cases}$$

因为它需要有 R 个 $f(x,y)$ 的值,所以称为 R 级的方法. 把系数 c_r,a_r 和 b_{rs} 列成表,或写成向量和矩阵的形式如下:

$$\begin{array}{c|ccc} a_1 & b_{11} & \cdots & b_{1R} \\ \vdots & \vdots & & \vdots \\ a_R & b_{R1} & \cdots & b_{RR} \\ \hline & c_1 & \cdots & c_R \end{array} \qquad \begin{array}{c|c} \boldsymbol{a} & \boldsymbol{B} \\ \hline & \boldsymbol{c}^{\mathrm{T}} \end{array} \qquad (10.4.23)$$

其中 $\boldsymbol{a}=(a_1,\cdots,a_R)^{\mathrm{T}}$,$\boldsymbol{c}^{\mathrm{T}}=(c_1,\cdots,c_R)$,$\boldsymbol{B}=(b_{ij})$. 系数之间的关系,根据相容性的要求,满足

$$\sum_{r=1}^{R} c_r = 1, \qquad (10.4.24)$$

$$a_r = \sum_{s=1}^{R} b_{rs} \quad (r = 1,\cdots,R). \qquad (10.4.25)$$

式(10.4.22)是 R 个函数 $K_1,\cdots,K_R$ 的非线性方程组,如果初值问题是 m 组向量函数 $\boldsymbol{y}$ 的初值问题(10.1.8),那么式(10.4.22)就是 mR 维的非线性方程组,所以隐式龙格-库塔方法的计算要比显式方法复杂很多,求解方程(10.4.22)的 $K_1,\cdots,K_R$ 可以用牛顿法等求解非线性方程组的方法. 如果用迭代法

$$K_r^{[t+1]} = f\left(x_n + a_r h, y_n + h\sum_{s=1}^{r-1} b_{rs} K_s^{[t+1]} + h\sum_{s=r}^{R} b_{rs} K_s^{[t]}\right)$$
$$(r = 1,\cdots,R), \qquad (10.4.26)$$

则若

$$h < \left\{L\left[\max_r \sum_{s=r}^{R} |b_{rs}| + \max_r \sum_{s=1}^{r-1} |b_{rs}|\right]\right\}^{-1},$$

迭代对于任意的初值 $K_1^{[0]}, K_2^{[0]}, \cdots, K_R^{[0]}$ 收敛,其中 L 是 $f(x,y)$ 对变量 y 的利普希茨常数.

如果 $\boldsymbol{B}$ 是一个严格下三角矩阵,即 $s \geqslant r$ 时有 $b_{rs}=0$,那么式(10.4.21)和式(10.4.22)是显式的龙格-库塔方法,此时 $K_1=f(x_n,y_n)$,由式(10.4.22)可逐步递推计算 $K_2, K_3, \cdots, K_R$. 如果$s>r$ 时有 $b_{rs}=0$,但 $\boldsymbol{B}$ 的对角元b_{rr}不全为 0,这时的方法称为**半显式的龙格-库塔方法**(semi-explicit Runge-Kutta methods). 若$b_{rr}\neq 0$,式(10.4.22)写成

$$K_r = f\left(x_n + a_r h, y_n + h\sum_{s=1}^{r-1} b_{rs}K_s + hb_{rr}K_r\right).$$

求 K_r 的迭代法可采用

$$K_r^{[t+1]} = f\left(x_n + a_r h, y_n + h\sum_{s=1}^{r-1} b_{rs}K_s + hb_{rr}K_r^{[t]}\right).$$

如果

$$h < \frac{1}{L\max\limits_r |b_{rr}|},$$

则迭代对任意初值 $K_r^{[0]}$ 收敛.

例 10.4.15　梯形方法:

$$\begin{cases} y_{n+1} = y_n + h\left(\dfrac{1}{2}K_1 + \dfrac{1}{2}K_2\right), \\ K_1 = f(x_n, y_n), \\ K_2 = f\left(x_n + h, y_n + \dfrac{h}{2}K_1 + \dfrac{h}{2}K_2\right). \end{cases}$$

它是一个二级二阶的隐式(半显式)龙格-库塔方法.

例 10.4.16　布特切尔方法(Butcher method):

$$
\begin{cases}
y_{n+1} = y_n + \dfrac{h}{6}(K_1 + 4K_2 + K_3), \\
K_1 = f(x_n, y_n), \\
K_2 = f\left(x_n + \dfrac{1}{2}h, y_n + \dfrac{h}{4}K_1 + \dfrac{h}{4}K_2\right), \\
K_3 = f(x_n + h, y_n + hK_2).
\end{cases}
$$

它是一个三级四阶的半显式方法.

对于 R 级的显式龙格-库塔方法，一般其阶 $p \leqslant R$，当 $R \leqslant 4$ 时可以有 $p=R$. 但是对 R 级的隐式龙格-库塔方法，其阶可以比显式方法高. 一般地 $p \leqslant 2R$，最高阶可以到 $p=2R$.

对于初值问题的积分形式(10.3.2)，如果用最高代数精确度的高斯-勒让德数值积分公式构造龙格-库塔方法. 可得 R 级的龙格-库塔方法的阶 $p=2R$.

例 10.4.17　隐式中点方法(implicit midpoint method)：

$$
y_{n+1} = y_n + hf\left(x_n + \frac{h}{2}, \frac{1}{2}(y_n + y_{n+1})\right),
$$

或

$$
\begin{cases}
y_{n+1} = y_n + hK_1, \\
K_1 = f\left(x_n + \dfrac{h}{2}, y_n + \dfrac{h}{2}K_1\right).
\end{cases}
$$

这是一个一级二阶的方法.

例 10.4.18　哈默-荷令斯沃思方法(Hammer-Hollingsworth method)

$$
\begin{cases}
y_{n+1} = y_n + \dfrac{h}{2}(K_1 + K_2), \\
K_1 = f\left(x_n + \left(\dfrac{1}{2} - \dfrac{\sqrt{3}}{6}\right)h, y_n + \dfrac{h}{4}K_1 + \left(\dfrac{1}{4} - \dfrac{\sqrt{3}}{6}\right)hK_2\right), \\
K_2 = f\left(x_n + \left(\dfrac{1}{2} + \dfrac{\sqrt{3}}{6}\right)h, y_n + \left(\dfrac{1}{4} + \dfrac{\sqrt{3}}{6}\right)hK_1 + \dfrac{h}{4}K_2\right).
\end{cases}
$$

这是一个二级四阶方法，按式(10.4.23)，系数表可写成

$\frac{1}{2}-\frac{\sqrt{3}}{6}$	$\frac{1}{4}$	$\frac{1}{4}-\frac{\sqrt{3}}{6}$
$\frac{1}{2}+\frac{\sqrt{3}}{6}$	$\frac{1}{4}+\frac{\sqrt{3}}{6}$	$\frac{1}{4}$
	$\frac{1}{2}$	$\frac{1}{2}$

例 10.4.19 一种三级六阶的隐式龙格-库塔方法. 系数表为

$\frac{1}{10}(5-\sqrt{15})$	$\frac{5}{36}$	$\frac{1}{45}\times(10-3\sqrt{15})$	$\frac{1}{180}\times(25-6\sqrt{15})$
$\frac{1}{2}$	$\frac{1}{72}\times(10+3\sqrt{15})$	$\frac{2}{9}$	$\frac{1}{72}\times(10-3\sqrt{15})$
$\frac{1}{10}(5+\sqrt{15})$	$\frac{1}{180}\times(25+6\sqrt{15})$	$\frac{1}{45}\times(10+3\sqrt{15})$	$\frac{5}{36}$
	$\frac{5}{18}$	$\frac{4}{9}$	$\frac{5}{18}$

利用拉道数值积分公式(见 4.6.1 节)构造隐式龙格-库塔方法，可得 R 级 $2R$-1 阶的拉道型方法.

例 10.4.20 两种 $R=2, p=3$ 拉道型的隐式龙格-库塔方法的系数表如下：

0	$\frac{1}{4}$	$-\frac{1}{4}$
$\frac{2}{3}$	$\frac{1}{4}$	$\frac{5}{12}$
	$\frac{1}{4}$	$\frac{3}{4}$

$\frac{1}{3}$	$\frac{5}{12}$	$-\frac{1}{12}$
1	$\frac{3}{4}$	$\frac{1}{4}$
	$\frac{3}{4}$	$\frac{1}{4}$

例 10.4.21 两种 $R=3, p=5$ 拉道型的隐式龙格-库塔方法的系数表如下：

$$
\begin{array}{c|ccc}
0 & \frac{1}{9} & \frac{-1-\sqrt{6}}{18} & \frac{-1+\sqrt{6}}{18} \\
\frac{6-\sqrt{6}}{10} & \frac{1}{9} & \frac{88+7\sqrt{6}}{360} & \frac{88-43\sqrt{6}}{360} \\
\frac{6+\sqrt{6}}{10} & \frac{1}{9} & \frac{88+43\sqrt{6}}{360} & \frac{88-7\sqrt{6}}{360} \\
\hline
 & \frac{1}{9} & \frac{16+\sqrt{6}}{36} & \frac{16-\sqrt{6}}{36}
\end{array}
$$

$$
\begin{array}{c|ccc}
\frac{4-\sqrt{6}}{10} & \frac{88-7\sqrt{6}}{360} & \frac{296-169\sqrt{6}}{1800} & \frac{-2+3\sqrt{6}}{225} \\
\frac{4+\sqrt{3}}{10} & \frac{296+169\sqrt{6}}{1800} & \frac{88+7\sqrt{6}}{360} & \frac{-2-3\sqrt{6}}{225} \\
1 & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \\
\hline
 & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9}
\end{array}
$$

利用洛巴托数值积分公式(见 4.6.2 节)构造隐式龙格-库塔方法,可得 R 级 $2R$-2 阶的洛巴托型方法.

例 10.4.22 三种 $R=2, p=2$ 的洛巴托型隐式龙格-库塔方法的系数表如下:

$$
\begin{array}{c|cc}
0 & 0 & 0 \\
1 & \frac{1}{2} & \frac{1}{2} \\
\hline
 & \frac{1}{2} & \frac{1}{2}
\end{array}
\qquad
\begin{array}{c|cc}
0 & \frac{1}{2} & 0 \\
1 & \frac{1}{2} & 0 \\
\hline
 & \frac{1}{2} & \frac{1}{2}
\end{array}
\qquad
\begin{array}{c|cc}
0 & \frac{1}{2} & -\frac{1}{2} \\
1 & \frac{1}{2} & \frac{1}{2} \\
\hline
 & \frac{1}{2} & \frac{1}{2}
\end{array}
$$

其中第一种方法即梯形方法.

例 10.4.23 三种 $R=3, p=4$ 的洛巴托型隐式龙格-库塔方法的系数表如下:

$$
\begin{array}{c|ccc}
0 & 0 & 0 & 0 \\
\frac{1}{2} & \frac{5}{24} & \frac{1}{3} & -\frac{1}{24} \\
1 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\
\hline
 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6}
\end{array}
\qquad
\begin{array}{c|ccc}
0 & \frac{1}{6} & -\frac{1}{6} & 0 \\
\frac{1}{2} & \frac{1}{6} & \frac{1}{3} & 0 \\
1 & \frac{1}{6} & \frac{5}{6} & 0 \\
\hline
 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6}
\end{array}
\qquad
\begin{array}{c|ccc}
0 & \frac{1}{6} & -\frac{1}{3} & \frac{1}{6} \\
\frac{1}{2} & \frac{1}{6} & \frac{5}{12} & -\frac{1}{12} \\
1 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\
\hline
 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6}
\end{array}
$$

例 10.4.24 三种 $R=4, p=6$ 的洛巴托型隐式龙格-库塔方法的系数表如下：

$$
\begin{array}{c|cccc}
0 & 0 & 0 & 0 & 0 \\
\frac{5-\sqrt{15}}{10} & \frac{11+\sqrt{5}}{120} & \frac{25-\sqrt{5}}{120} & \frac{25-13\sqrt{5}}{120} & \frac{-1+\sqrt{5}}{120} \\
\frac{5+\sqrt{15}}{10} & \frac{11-\sqrt{5}}{120} & \frac{25+13\sqrt{5}}{120} & \frac{25+\sqrt{5}}{120} & \frac{-1-\sqrt{5}}{120} \\
1 & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} \\
\hline
 & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12}
\end{array}
$$

$$
\begin{array}{c|cccc}
0 & \frac{1}{12} & \frac{-1-\sqrt{5}}{24} & \frac{-1+\sqrt{5}}{24} & 0 \\
\frac{5-\sqrt{5}}{10} & \frac{1}{12} & \frac{25+\sqrt{5}}{120} & \frac{25-13\sqrt{5}}{120} & 0 \\
\frac{5+\sqrt{5}}{10} & \frac{1}{12} & \frac{25+13\sqrt{5}}{120} & \frac{25-\sqrt{5}}{120} & 0 \\
1 & \frac{1}{12} & \frac{11-\sqrt{5}}{24} & \frac{11+\sqrt{5}}{24} & 0 \\
\hline
 & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12}
\end{array}
$$

$$
\begin{array}{c|cccc}
0 & \frac{1}{12} & \frac{-\sqrt{5}}{12} & \frac{\sqrt{5}}{12} & -\frac{1}{12} \\
\frac{5-\sqrt{5}}{10} & \frac{1}{12} & \frac{1}{4} & \frac{10-7\sqrt{5}}{60} & \frac{\sqrt{5}}{60} \\
\frac{5+\sqrt{5}}{10} & \frac{1}{12} & \frac{10+7\sqrt{5}}{60} & \frac{1}{4} & -\frac{\sqrt{5}}{60} \\
1 & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} \\
\hline
 & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12}
\end{array}
$$

10.4.8 单步法的绝对稳定性

用一种数值方法求解初值问题，计算过程总会产生误差. 设在某一步有舍入误差，它会影响到以后各步的计算，如果它引起在计算过程中误差成为无界的，那就是数值不稳定现象，实际计算中应当避免. 这种稳定的概念与所采用的计算方法有关，也与方程中的函数 $f(x,y)$ 有关. 为了只考察数值方法本身，通常检验数值方法用于试验方程的稳定性. 试验方程为

$$\frac{\mathrm{d}y}{\mathrm{d}x}=\lambda y \tag{10.4.27}$$

选择方程(10.4.27)为试验方程，是因为它比较简单，同时任何一个微分方程可以局部线性化为这种形式. 例如，在点 (a,b) 的邻域有

$$f(x,y)=f(a,b)+(x-a)\frac{\partial f}{\partial x}(a,b)+(y-b)\frac{\partial f}{\partial y}(a,b)+\cdots$$

略去高阶项并做适当变量置换，可将方程 $y'=f(x,y)$ 化为方程(10.4.27)的形式，其中 λ 对应$\frac{\partial f}{\partial y}$. 如果是 m 个方程的方程组 $\mathbf{y}'=f(x,\mathbf{y})$，线性化后得线性方程组 $\mathbf{y}'=\mathbf{A}\mathbf{y}$，$\mathbf{A}$ 为 $m\times m$ 的雅可比矩阵$\left(\frac{\partial f_i}{\partial y_j}\right)$. 若 $\mathbf{A}$ 有 m 个不同的特征值 $\lambda_1,\cdots,\lambda_m$，可做正交相似变换化 $\mathbf{A}$ 为对角阵 $\mathrm{diag}(\lambda_1,\cdots,\lambda_m)$，得到 m 个方程 $y_i'=\lambda_i y_i\,(i=1,\cdots,m)$. 所以，考虑方程(10.4.27)中 λ 为复数.

一个单步法用于试验方程(10.4.27)，从 y_n 计算一步得到

$$y_{n+1}=E(\lambda h)y_n, \tag{10.4.28}$$

其中 $E(\lambda h)$ 依赖于所选的方法，而且 $E(\lambda h)$ 是 $\mathrm{e}^{\lambda h}$ 的一个近似值. 如果 y_n 的计算有误差 ε，它会引起 y_{n+1} 的误差 $E(\lambda h)\varepsilon$. 如果在 y_0 有误差 ε_0，它引起 y_{n+1} 的误差 $(E(\lambda h))^{n+1}\varepsilon_0$.

定义 10.4.25 对某一种方法，如果式(10.4.28)中 $E(\lambda h)$ 满

足$|E(\lambda h)|<1$，称此方法**绝对稳定**(absolutely stable)，在μ复平面上复变量μ满足$|E(\mu)|<1$的区域，称为此方法的**绝对稳定区域**(region of absolute stability)，它与实轴的交称为**绝对稳定区间**(interval of absolute stability).

例 10.4.26 将欧拉方法用于试验方程(10.4.27)，有

$$y_{n+1}=y_n+\lambda h y_n=(1+\mu)y_n.$$

其中$\mu=\lambda h, E(\mu)=1+\mu$. $E(\mu)$是e^{μ}的一阶泰勒近似. 当$|1+\mu|<1$时，欧拉方法是绝对稳定的. 欧拉方法的绝对稳定区域是一个以-1为中心的单位圆，如图10.2所示，它的绝对稳定区间是$(-2,0)$.

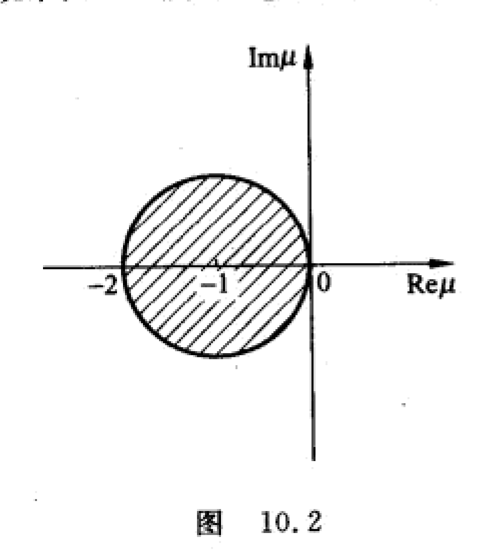


图 10.2

当λ为负实数时，取$h\leqslant-\dfrac{2}{\lambda}$，欧拉方法是稳定的.

例 10.4.27 用欧拉方法计算

$$\begin{cases}\dfrac{\mathrm{d}y}{\mathrm{d}x}=-1000(y-x^2)+2x \quad (0\leqslant x\leqslant 1),\\ y(0)=0.\end{cases}$$

解 设计算机有12位有效数字的精确度，取$h=10^{-m}$ $(m=0,1,2,\cdots)$，从$x=0$计算到$x=1$，结果如下：

h	$y(1)$的计算值
1	0
0.1	$0.90438207503\times10^{16}$
0.01	溢出
0.001	0.99999900001
0.0001	0.99999990000
0.00001	0.99999998997

表中"溢出"表示超过了机器能容纳的最大数.本例相当于$\lambda=-1000$,当$h=0.01$时,$1+\lambda h=-9$,即每个误差绝对值放大9倍,而计算$x=1$有100步,误差约放大10^{100}倍,显然得不到合理的结果.但是只要h适当小,使$|1+\lambda h|<1$.计算结果就合理.

例 10.4.28 将四阶经典龙格-库塔方法用于解试验方程(10.4.27),可得到

$$y_{n+1}=\left(1+\lambda h+\frac{(\lambda h)^2}{2}+\frac{(\lambda h)^3}{6}+\frac{(\lambda h)^4}{24}\right)y_n.$$

令$\mu=\lambda h$,有$E(\mu)=1+\mu+\frac{\mu^2}{2}+\frac{\mu^3}{6}+\frac{\mu^4}{24}$,$E(\mu)$是$e^\mu$的四阶泰勒近似.经典龙格-库塔方法的绝对稳定区域为$|E(\mu)|<1$,即在μ平面上,由曲线

$$1+\mu+\frac{\mu^2}{2}+\frac{\mu^3}{6}+\frac{\mu^4}{24}=e^{i\theta}$$

所围成的区域,见图10.3.其中θ是复数的辐角.

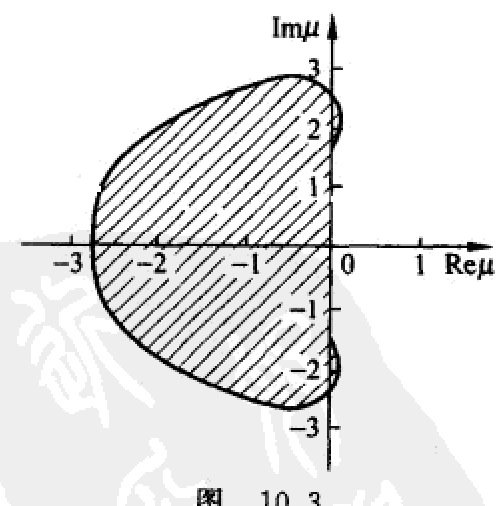


图 10.3

正数 h 都有 $\lambda h\in(-\infty,0)$，即步长 h 不必受限制.

例 10.4.31 将二级二阶的梯形方法(10.3.4)和一级二阶的隐式中点方法(见例 10.4.17)用于试验方程，都得到

$$E(\mu)=\frac{1+\frac{\mu}{2}}{1-\frac{\mu}{2}},$$

这是函数 e^{μ} 的帕德(1,1)有理逼近. 因为对左半平面都有 $|E(\mu)|<1$，所以方法绝对稳定区域是 μ 平面的左半平面，绝对稳定区间是 $(-\infty,0)$. 而二级二阶显式龙格-库塔方法的绝对稳定区间是 $(-2,0)$. 从绝对稳定的意义看来，隐式方法比同阶的显式方法要优越.

例 10.4.32 将二级四阶的哈默-荷令斯沃思方法(见例 10.4.18)用于试验方程，可得

$$E(\mu)=\frac{1+\frac{\mu}{2}+\frac{\mu^2}{12}}{1-\frac{\mu}{2}+\frac{\mu^2}{12}},$$

这是函数 e^{μ} 的帕德(2,2)有理逼近，方法的绝对稳定区间是 $(-\infty,0)$，优于四级四阶的显式龙格-库塔方法，后者绝对稳定区间是 $(-2.78,0)$.

例 10.4.33 将半显式的三级四阶布特切尔方法(见例 10.4.16)用于试验方程，可得

$$E(\mu)=\frac{1+\frac{3}{4}\mu+\frac{1}{4}\mu^2+\frac{1}{24}\mu^3}{1-\frac{1}{4}\mu},$$

这是函数 e^{μ} 的帕德(3,1)逼近，方法的绝对稳定区间是 $(-5.41,0)$，优于四级四阶的显式龙格-库塔方法.

定理 10.4.34 设隐式单步法用于试验方程，得到 $y_{n+1}=E(\mu)y_n$，其中 $\mu=\lambda h$. 则

(1) 由高斯-勒让德积分公式构造的 R 级 $2R$ 阶的隐式龙格

例 10.4.29 用经典龙格-库塔方法计算

$$\begin{cases} \dfrac{dy}{dx} = -20y \quad (0 \leqslant x \leqslant 1), \\ y(0) = 1, \end{cases}$$

取 $h=0.1$ 及 $h=0.2$.

解 本例 $\lambda=-20$, $\mu=\lambda h$ 分别为 -2 和 -4. 前者属于绝对稳定区间,后者则不属于. 计算结果误差如下表.

x_n	$h=0.1$ 时误差	$h=0.2$ 时误差
0.0	0	0
0.2	−0.092795	4.98
0.4	−0.012010	25.0
0.6	−0.001366	125.0
0.8	−0.000152	625.0
1.0	−0.000017	3125.0

$R=1,2,3,4$ 时 R 级 R 阶的显式龙格-库塔方法的绝对稳定区间如表 10.7:

表 10.7

R	$E(\mu)$	绝对稳定区间
1	$1+\mu$	$(-2,0)$
2	$1+\mu+\dfrac{\mu^2}{2}$	$(-2,0)$
3	$1+\mu+\dfrac{\mu^2}{2}+\dfrac{\mu^3}{6}$	$(-2.51,0)$
4	$1+\mu+\dfrac{\mu^2}{2}+\dfrac{\mu^3}{6}+\dfrac{\mu^4}{24}$	$(-2.78,0)$

例 10.4.30 将隐式欧拉方法(10.3.3)用于试验方程,令 $\mu=\lambda h$,可得

$$E(\mu) = \frac{1}{1-\mu},$$

这是指数函数 e^{μ} 的帕德(Padé)(0,1)有理逼近(见 3.1.2 节),方法的绝对稳定区间是 $(-\infty,0)$. 当 λ 是小于 0 的实数时,对任意的

库塔方法(例 10.4.17～例 10.4.19 等),$E(\mu)$为函数 e^{μ} 的帕德(R,R)有理逼近.

(2) 由拉道积分公式构造的 R 级 $2R-1$ 阶的隐式龙格-库塔方法(例 10.4.20、例 10.4.21 等). $E(\mu)$为函数 e^{μ} 的帕德$(R-1,R)$有理逼近.

(3) 由洛巴托积分公式构造的 R 级 $2R-2$ 阶的隐式龙格-库塔方法(例 10.4.22～例 10.4.24 等),$E(\mu)$为函数 e^{μ} 的帕德$(R-1,R-1)$或$(R-2,R)$有理逼近.

帕德(m,n)逼近的有理式,见 3.12 节的 e^{x} 帕德表.

10.5 线性多步法

10.5.1 线性多步法的一般概念

如式(10.1.15)所示,解初值问题(10.1.4),(10.1.5)的 k 步线性多步法一般形式为

$$\begin{aligned} y_{n+k} = & -\alpha_0 y_n - \alpha_1 y_{n+1} - \cdots - \alpha_{k-1} y_{n+k-1} + \\ & h(\beta_0 f_n + \beta_1 f_{n+1} + \cdots + \beta_k f_{n+k}). \end{aligned} \tag{10.5.1}$$

可以由泰勒展开式方法或数值积分公式方法来确定一种线性多步法的系数$\{\alpha_j,\beta_j\}$.

对应于一种线性多步法(10.5.1),可以定义算子

$$\mathscr{L}[y(x);h] = \sum_{i=0}^{k}[\alpha_i y(x+ih) - h\beta_i y'(x+ih)], \tag{10.5.2}$$

其中 $\alpha_k=1$,$y(x)$是$[x_0,b]$上任意的连续可微函数.如果将 $y(x+ih)$及 $y'(x+ih)(i=0,1,\cdots,k)$按泰勒公式展开:

$$y(x+ih) = y(x) + \frac{ih}{1!}y'(x) + \frac{(ih)^2}{2!}y''(x) + \cdots,$$

$$y'(x+ih)=y'(x)+\frac{ih}{1!}y''(x)+\frac{(ih)^2}{2!}y'''(x)+\cdots,$$

代入式(10.5.2),可整理成

$$\mathscr{L}[y(x);h]=c_0y(x)+c_1hy'(x)+\cdots+c_qh^qy^{(q)}(x)+\cdots,\tag{10.5.3}$$

其中

$$\begin{cases}c_0=\alpha_0+\alpha_1+\alpha_2+\cdots+\alpha_k,\\ c_1=\alpha_1+2\alpha_2+\cdots+k\alpha_k-(\beta_0+\beta_1+\beta_2+\cdots+\beta_k)\\ c_q=\dfrac{1}{q!}(\alpha_1+2^q\alpha_2+\cdots+k^q\alpha_k)-\\ \qquad\dfrac{1}{(q-1)!}(\beta_1+2^{q-1}\beta_2+\cdots+k^{q-1}\beta_k)\quad(q=2,3,\cdots)\end{cases}\tag{10.5.4}$$

定义 10.5.1 若在式(10.5.3)中,$c_0=c_1=\cdots=c_p=0$,$c_{p+1}\neq0$,则称算子(10.5.2)对应的线性多步法(10.5.1)是 p 阶的. 若 $p\geqslant1$,称线性多步法(10.5.1)与初值问题(10.1.4),(10.1.5)**相容**.

定理 10.5.2 k 步线性多步法(10.5.1)与初值问题(10.1.4),(10.1.5)相容的充分必要条件是:

$$\sum_{i=0}^{k}\alpha_i=0,\quad\sum_{i=1}^{k}i\alpha_i=\sum_{i=0}^{k}\beta_i.\tag{10.5.5}$$

实际使用的方法应该是相容的,所以要寻找系数 $\alpha_0,\cdots,\alpha_k$,$\beta_0,\cdots,\beta_k$,使得 $c_0=c_1=\cdots=c_p=0$,$c_{p+1}\neq0$(其中 $p\geqslant1$). 得到的系数代入式(10.5.1),即可构成一个 p 阶的线性 k 步法. 当 $\beta_k=0$ 时方法是显式的,当 $\beta_k\neq0$ 时方法是隐式的.

定义 10.5.3 设 $y(x)$ 是初值问题(10.1.4),(10.1.5)的准确解,则

$$e_{n+k}=y(x_{n+k})-y_{n+k}$$

称为方法(10.5.1)在 x_{n+k} 的**整体截断误差**.

$$T_{n+k}=\mathscr{L}[y(x_n);h]$$

称为方法(10.5.1)在 x_{n+k} 的**局部截断误差**.

据定义 10.5.3,若线性多步法(10.5.1)是 p 阶的方法,则其局部截断误差

$$T_{n+k}=c_{p+1}h^{p+1}y^{(p+1)}(x_n)+c_{p+2}h^{p+2}y^{(p+2)}(x_n)+\cdots. \tag{10.5.6}$$

定理 10.5.4 假设计算 $y(x_{n+k})$ 前 k 步所得离散解是准确的,即设 $y_{n+i}=y(x_{n+i})(i=0,1,\cdots,k-1)$,则对于显式的线性多步法($\beta_k\neq 0$),有

$$T_{n+k}=y(x_{n+k})-y_{n+k}.$$

而对于 p 阶的隐式多步法($\beta_k\neq 0$),有

$$y(x_{n+k})-y_{n+k}=c_{p+1}h^{p+1}y^{(p+1)}(x_n)+O(h^{p+2}).$$

即对于隐式的线性多步法,设前 k 步离散解准确时,局部截断误差(10.5.6)与 $y(x_{n+k})-y_{n+k}$ 的展开式首项相同.

定理 10.5.5 p 阶线性多步法局部截断误差 T_{n+k} 的首项 $c_{p+1}h^{p+1}y^{(p+1)}(x_n)$ 称为方法的**主局部截断误差**. c_{p+1} 称**误差常数**.

例如,欧拉方法 $y_{n+1}=y_n+hf(x_n,y_n)$ 是单步的显式方法,相当于在式(10.5.1)中,$k=1,\alpha_0=-1,\beta_0=1,\beta_1=0$. 由式(10.5.4),有 $c_0=c_1=0,c_2=\dfrac{1}{2}$. 所以欧拉方法是一阶方法,其主局部截断误差 $T_{n+1}=\dfrac{1}{2}h^2y''(x_n)$. 这里的分析和 10.3 节的分析是一致的.

在区间$[x_{n+k-l},x_{n+k}]$上,初值问题(10.1.4),(10.1.5)的解满足

$$y(x_{n+k})=y(x_{n+k-l})+\int_{x_{n+k-l}}^{x_{n+k}}f(t,y(t))\mathrm{d}t. \tag{10.5.7}$$

如果用各种数值积分公式近似式(10.5.7)右端的积分式,就可得到多种计算 $y(x_{n+k})$ 的近似值 y_{n+k} 的计算方法. 这些方法称为**基于数值积分公式的方法**.

例如,在式(10.5.7)中令 $k=l=1$,右端积分式分别用左矩形

公式、右矩形公式和梯形数值积分公式近似，就得到单步的欧拉方法、隐式欧拉方法和梯形方法.

例 10.5.6　辛普森方法(Simpson method)在式(10.5.7)中取 $k=l=2$，该式右端积分式用节点为 x_n, x_{n+1}, x_{n+2} 的辛普森数值积分公式近似，再用 y_{n+j} 代替 $y(x_{n+j})(j=0,1,2)$. 就得到

$$y_{n+2}=y_n+\frac{h}{3}(f_n+4f_{n+1}+f_{n+2}).$$

这就是辛普森方法，它的局部截断误差为

$$T_{n+2}=-\frac{1}{90}h^5y^{(5)}(x_n)+O(h^6).$$

方法是二步四阶的方法.

例 10.5.7　米尔恩方法(Milne method)

在式(10.5.7)中取 $k=l=4$，该式右端函数用节点为 $x_{n+1}, x_{n+2}, x_{n+3}$ 的二次插值多项式近似，代入积分式后，再用 y_{n+j} 代替 $y(x_{n+j})(j=0,1,2,3,4)$，就得到

$$y_{n+4}=y_n+\frac{4h}{3}(2f_{n+1}-f_{n+2}+2f_{n+3}).$$

这就是米尔恩方法，它的局部截断误差为

$$T_{n+4}=\frac{14}{45}h^5y^{(5)}(x_n)+O(h^6).$$

方法是四步四阶的方法.

10.5.2　亚当斯方法

在式(10.5.7)中取 $l=1$，得到

$$y(x_{n+k})=y(x_{n+k-1})+\int_{x_{n+k-1}}^{x_{n+k}}f(t,y(t))\mathrm{d}t, \quad (10.5.8)$$

再用数值积分公式，这类方法称**亚当斯方法**(Adams methods).

如果式(10.5.8)中 $f(t,y(t))$ 用 $r+1$ 个节点 $\{x_{n+k-1-j}\}_{j=0}^{r}$ 的插值多项式代替，代入式(10.5.8)做积分，以近似值 y_{n+j} 代替

$y(x_{n+j})$，就得到如下的**亚当斯-巴什福思方法**(Adams-Bashforth methods).

$$y_{n+k} = y_{n+k-1} + h(\rho_{r0} f_{n+k-1} + \rho_{r1} f_{n+k-2} + \cdots + \rho_{rr} f_{n+k-1-r}), \tag{10.5.9}$$

其中

$$\rho_{rj} = \frac{1}{h}\int_{x_{n+k-1}}^{x_{n+k}} l_j(x)\mathrm{d}x \quad (j = 0,1,\cdots,r),$$

$l_j(x)$是节点 $x_{n+k-1-j}$ 上的插值基函数. 方法(10.5.9)是 $r+1$ 步的显式公式，也称**亚当斯显式方法**，其中的系数、阶和误差常数 c_{p+1} 如表 10.8 所示.

表 10.8

r	$r+1$(步)	p(阶)	ρ_{r0}	ρ_{r1}	ρ_{r2}	ρ_{r3}	ρ_{r4}	c_{p+1}
0	1	1	1					$\frac{1}{2}$
1	2	2	$\frac{3}{2}$	$-\frac{1}{2}$				$\frac{5}{12}$
2	3	3	$\frac{23}{12}$	$-\frac{16}{12}$	$\frac{5}{12}$			$\frac{3}{8}$
3	4	4	$\frac{55}{24}$	$-\frac{59}{24}$	$\frac{37}{24}$	$-\frac{9}{24}$		$\frac{251}{720}$
4	5	5	$\frac{1901}{720}$	$-\frac{2774}{720}$	$\frac{2616}{720}$	$-\frac{1274}{720}$	$\frac{251}{720}$	$\frac{95}{288}$

$r=0$ 时，亚当斯显式方法就是欧拉方法. $r=3$ 时，取 $k=r+1$，公式(10.5.9)为

$$y_{n+4} = y_{n+3} + \frac{h}{24}(55f_{n+3} - 59f_{n+2} + 37f_{n+1} - 9f_n). \tag{10.5.10}$$

它是四步四阶的方法，局部截断误差

$$T_{n+4} = \frac{251}{720}h^5 y^{(5)}(x_n) + O(h^6).$$

如果在公式(10.5.8)中，$f(t,y(t))$用 $r+1$ 个节点 $\{x_{n+k-j}\}_{j=0}^{r}$的插值多项式代替，代入式(10.5.8)做积分，以近似值 y_{n+j}代替 $y(x_{n+j})$，就得到如下的**亚当斯-莫尔顿方法**(Adams-Moulton methods).

$$y_{n+k} = y_{n+k-1} + h(\rho_{r0} f_{n+k} + \rho_{r1} f_{n+k-1} + \cdots + \rho_{rr} f_{n+k-r}), \tag{10.5.11}$$

其中

$$\rho_{rj} = \frac{1}{h}\int_{x_{n+k-1}}^{x_{n+k}} l_j(x)\mathrm{d}x \quad (j = 0,1,\cdots,r).$$

$l_j(x)$是节点 x_{n+k-j}上的插值基函数. 当 $r\geqslant 1$ 时，方法(10.5.11)是 r 步的隐式方法. 式(10.5.11)也称**亚当斯隐式方法**，其中的系数、阶和误差常数 c_{p+1}如表 10.9 所示.

表 10.9

r	步	p(阶)	ρ_{r0}	ρ_{r1}	ρ_{r2}	ρ_{r3}	ρ_{r4}	c_{p+1}
0	1	1	1					$-\frac{1}{2}$
1	1	2	$\frac{1}{2}$	$\frac{1}{2}$				$-\frac{1}{12}$
2	2	3	$\frac{5}{12}$	$\frac{8}{12}$	$-\frac{1}{12}$			$-\frac{1}{24}$
3	3	4	$\frac{9}{24}$	$\frac{19}{24}$	$-\frac{5}{24}$	$\frac{1}{24}$		$-\frac{19}{720}$
4	4	5	$\frac{251}{720}$	$\frac{646}{720}$	$-\frac{264}{720}$	$\frac{106}{720}$	$-\frac{19}{720}$	$-\frac{3}{160}$

$r=0$ 时，亚当斯隐式方法就是隐式欧拉方法. $r=1$ 时是梯形方法. $r=3$ 时，取 $k=4$，公式(10.5.11)为

$$y_{n+4} = y_{n+3} + \frac{h}{24}(9f_{n+4} + 19f_{n+3} - 5f_{n+2} + f_{n+1}). \tag{10.5.12}$$

它是三步四阶的方法，局部截断误差

$$T_{n+4}=-\frac{19}{720}h^5y^{(5)}(x_n)+O(h^6).$$

例 10.5.8 分别用四阶的亚当斯显式方法(10.5.10)和隐式方法(10.5.12)解初值问题

$$\begin{cases}\dfrac{\mathrm{d}y}{\mathrm{d}x}=-y+x+1 \quad (0\leqslant x\leqslant 1),\\ y(0)=1,\end{cases}$$

步长取为 $h=0.1$.

解 初值问题的准确解为 $y(x)=\mathrm{e}^{-x}+x$. 计算结果及误差(取绝对值)如表 10.10. 从表中可见四阶隐式方法误差要比四阶显式方法的误差要小些.

表 10.10

x_n	亚当斯显式方法		亚当斯隐式方法	
	y_n	误 差	y_n	误 差
0.3			1.04081801	2.1×10^{-7}
0.4	1.07032292	2.87×10^{-6}	1.07031966	3.9×10^{-7}
0.5	1.10653548	4.82×10^{-6}	1.10653014	5.2×10^{-7}
0.6	1.14881841	6.77×10^{-6}	1.14881101	6.3×10^{-7}
0.7	1.19659339	8.09×10^{-6}	1.19658459	7.1×10^{-7}
0.8	1.24933816	9.19×10^{-6}	1.24932819	7.7×10^{-7}
0.9	1.30657961	9.95×10^{-6}	1.30656885	8.1×10^{-7}
1.0	1.36788996	1.05×10^{-5}	1.36787860	8.4×10^{-7}

10.5.3 尼斯特龙方法

在式(10.5.7)中取 $l=2$ 的方法称为尼斯特龙方法(Nyström methods). 如果式(10.5.7)中积分式的被积函数用节点 x_{n+k-1}, x_{n+k-2}, $\cdots$, $x_{n+k-(r+1)}$ 上的插值多项式代替,得到显式方法

$$y_{n+k}=y_{n+k-2}+h(\rho_{r0}f_{n+k-1}+\rho_{r1}f_{n+k-2}+\cdots+\rho_{rr}f_{n+k-1-r}). \tag{10.5.13}$$

其中

$$\rho_{rj} = \int_{x_{n+k-2}}^{x_{n+k}} l_j(x)\mathrm{d}x \quad (j = 0,1,\cdots,r),$$

$l_j(x)$是对应节点上的插值基函数. 特别取 $r=0$ 时,得 $\rho_{00}=2$,方法为

$$y_{n+k} = y_{n+k-2} + 2hf_{n+k-1},$$

若取 $k=2$,则写成

$$y_{n+2} = y_n + 2hf_{n+1}.$$

这就是所谓的**中点方法**.

如果在 $l=2$ 的式(10.5.7)中被积函数用节点 x_{n+k}, $x_{n+k-1},\cdots,x_{n+k-r}$上的插值多项式代替,得到隐式方法

$$y_{n+k} = y_{n+k-2} + h(\rho_{r0}f_{n+k} + \rho_{r1}f_{n+k-1} + \cdots + \rho_{rr}f_{n+k-r}). \tag{10.5.14}$$

其中系数也是对应的插值基函数的积分. 特别取 $r=2$ 时就是例 10.5.6 的辛普森方法.

10.5.4 汉明方法

汉明方法(Hamming method)是一种四阶隐式方法,可以利用泰勒展开式的方法来推导,它的计算公式为

$$y_{n+3} = \frac{1}{8}(9y_{n+2} - y_n) + \frac{3h}{8}(f_{n+3} + 2f_{n+2} - f_{n+1}), \tag{10.5.15}$$

方法的局部截断误差为$-\frac{h^5}{40}y^{(5)}(x_n)+O(h^6)$.

10.5.5 线性多步法的收敛性

如果将线性多步法(10.5.1)用到试验方程 $y'=\lambda y$,就得到

$$\sum_{j=0}^{k}(\alpha_j - h\lambda\beta_j)y_{n+j} = 0. \tag{10.5.16}$$

它是一个常系数 k 阶差分方程. 令 $y_{n+j}=r^j$ 代入式(10.5.16),就得到此差分方程的特征方程

$$\pi(r,h)=0,$$

其中

$$\pi(r,h)=\rho(r)-h\lambda\sigma(r),$$

$$\rho(r)=\sum_{j=0}^{k}\alpha_j r^j,\quad \sigma(r)=\sum_{j=0}^{k}\beta_j r^j. \tag{10.5.17}$$

由线性多步法(10.5.1)完全确定了多项式 $\rho(r)$ 和 $\sigma(\lambda)$，反之也成立.

根据定理 10.5.2，线性多步法(10.5.1)与初值问题(10.1.4)，(10.1.5)相容的充分必要条件是：

$$\rho(1)=0,\quad \rho'(1)=\sigma(1). \tag{10.5.18}$$

如果用线性多步法(10.5.1)及 k 个初始离散条件解初值问题，则数值解由下列差分方程给出：

$$\begin{cases}\sum_{j=0}^{k}\alpha_j y_{n+j}=h\sum_{j=0}^{k}\beta_j f_{n+j},\\ y_\mu=\eta_\mu(h)\quad(\mu=0,1,\cdots,k-1).\end{cases} \tag{10.5.19}$$

定义 10.5.9 如果对所有 $f(x,y)$ 满足对 y 的利普希茨条件的初值问题(10.1.4)，(10.1.5)，用方法(10.5.19)计算，当初始条件 $y_\mu=\eta_\mu(h)$ 满足条件

$$\lim_{h\to 0}\eta_\mu(h)=y_0\quad(\mu=0,1,\cdots,k-1)$$

时，式(10.5.19)的解 $\{y_n\}$ 有：对一切固定的 $x\in[x_0,b]$，$x=x_0+nh$，

$$\lim_{h\to 0}y_n=y(x),$$

则称线性多步法(10.5.19)是**收敛**的.

定理 10.5.10 若线性多步法(10.5.1)是收敛的，则它与初值问题(10.1.4)，(10.1.5)相容.

定义 10.5.11 如果线性多步法(10.5.1)对应的多项式 $\rho(r)$ 的 k 个根 $r_1,r_2,\cdots,r_k$ 满足

$$|r_l|\leqslant 1,\quad l=1,2,\cdots,k,$$

且等号只对单根成立，就称此方法满足**根条件**(root condition).

根条件的意义在于：$\rho(r)$的根都在复平面的单位圆上或圆内，如果是在单位圆上，则该根为单根.

定理 10.5.12 若线性多步法(10.5.1)与初值问题(10.1.4)，(10.1.5)相容，则方法(10.5.19)收敛的充分必要条件是它满足根条件.

10.5.6 线性多步法的稳定性

设计算方法(10.5.19)的初始条件有了扰动，且差分方程右端也有扰动，方法写成

$$\begin{cases}\sum_{j=0}^{k}\alpha_j z_{n+j} = h\left[\sum_{j=0}^{k}\beta_j f(x_{n+j}, z_{n+j}) + \delta_{n+k}\right] \\ z_\mu = \eta_\mu(h) + \delta_\mu \quad (\mu = 0,1,\cdots,k-1)\end{cases} \tag{10.5.20}$$

其中$\{\delta_n\}_{n=0}^N$是方法的扰动($n=0,1,\cdots,k-1$ 是初始条件的扰动，$n=k,\cdots,N$ 是差分方程右端的扰动)，$\{z_n\}_{n=0}^N$是扰动后的方法的数值解结果，这里 $N=\left[\frac{b-x_0}{h}\right]$.

定义 10.5.13 对所有 $f(x,y)$满足对 y 利普希茨条件的初值问题(10.1.4)，(10.1.5)，设$\{\delta_n\}_{n=0}^N$和$\{\delta_n^*\}_{n=0}^N$是线性多步法(10.5.19)的两个扰动，$\{z_n\}_{n=0}^N$和$\{z_n^*\}_{n=0}^N$是扰动后的解. 如果存在不依赖 h 的常数 C 和 $h_0>0$，使得对任意步长 $h\in(0,h_0]$，当

$$|\delta_n - \delta_n^*| \leqslant \varepsilon \quad (n = 0,1,\cdots,N)$$

时有

$$|z_n - z_n^*| \leqslant C\varepsilon \quad (n = 0,1,\cdots,N).$$

则称线性多步法(10.5.1)是**稳定的**.

以上定义的稳定性是考虑在 $h\to 0$ 情况下的稳定问题，也称**零稳定**(zero-stable).

定理 10.5.14 线性多步法(10.5.1)是零稳定的充分必要条件是它满足根条件.

例如,例 10.5.6 的辛普森方法,有

$$\pi(r,h)=\left(1-\frac{\lambda h}{3}\right)r^2-\frac{4\lambda h}{3}r-\left(1+\frac{\lambda h}{3}\right),$$

$$\rho(r)=\pi(r,0)=r^2-1.$$

$\pi(r,h)$的根

$$r_1(h)=1+\lambda h+O(h^2),$$

$$r_2(h)=-1+\frac{1}{3}\lambda h+O(h^2).$$

而 $\rho(r)$的根 $r_1=1,r_2=-1$. 所以方法满足根条件,是零稳定的. 方法是四阶的方法,根据定理 10.5.12 知是收敛的. 但是如果解方程 $y'=\lambda y$,若 λ 为正数,且 h 充分小有$|r_1(h)|>1$,若 λ 为负数,且 h 充分小有$|r_2(h)|>1$. 将会不满足下面讨论的绝对稳定性. 所以此方法一般不推荐使用.

10.5.7 线性多步法的绝对稳定性

绝对稳定性是讨论固定 h 情形的稳定性. 将线性多步法用于试验方程 $y'=\lambda y$,得到差分方程(10.5.16). 它的特征方程是

$$\rho(r)-\mu\sigma(r)=0, \tag{10.5.21}$$

其中 $\mu=\lambda h$. 它的根 $r_1,r_2,\cdots,r_k$ 如果都是单根,则差分方程(10.5.16)的解为

$$y_n=\sum_{j=1}^{k}B_jr_j^n.$$

如果有一个根 r_j 是 q 重的,则和式中会产生

$$[B_{j1}+nB_{j2}+n(n-1)B_{j3}+\cdots+n(n-1)\cdots(n-q+2)B_{jq}]r_j^n$$

的项. 用多步法解试验方程,y_n 的误差也满足差分方程(10.5.16). 所以,若$|r_j|<1$,当 n 增大时误差是有界的,(但若$|r_j|=1$,n 增大时误差会变得无界),所以不考虑等号情形.

定义 10.5.15 对于给定的 μ,如果特征方程(10.5.21)的根 r_j 满足$|r_j|<1(j=1,2,\cdots,k)$,则称线性多步法(10.5.1)是绝对

稳定的.在复平面 μ 中,使方法(10.5.1)绝对稳定的区域,称为方法的**绝对稳定区域**,它和实轴的交集称为方法的**绝对稳定区间**.而 $|r_j|<|r_1|(j=2,3,\cdots,k)$ 时称方法**相对稳定**(relatively stable).

例 10.5.16 亚当斯显式方法和隐式方法的绝对稳定区间如表 10.11.

表 10.11

k(步)	亚当斯显式方法		亚当斯隐式方法	
	p(阶)	绝对稳定区间	p(阶)	绝对稳定区间
1	1	$(-2,0)$	2	$(-\infty,0)$
2	2	$(-1,0)$	3	$(-6,0)$
3	3	$\left(-\frac{6}{11},0\right)$	4	$(-3,0)$
4	4	$\left(-\frac{3}{10},0\right)$	5	$\left(-\frac{90}{49},0\right)$

亚当斯显式方法和隐式方法的绝对稳定区域分别如图 10.4 和图 10.5 所示.

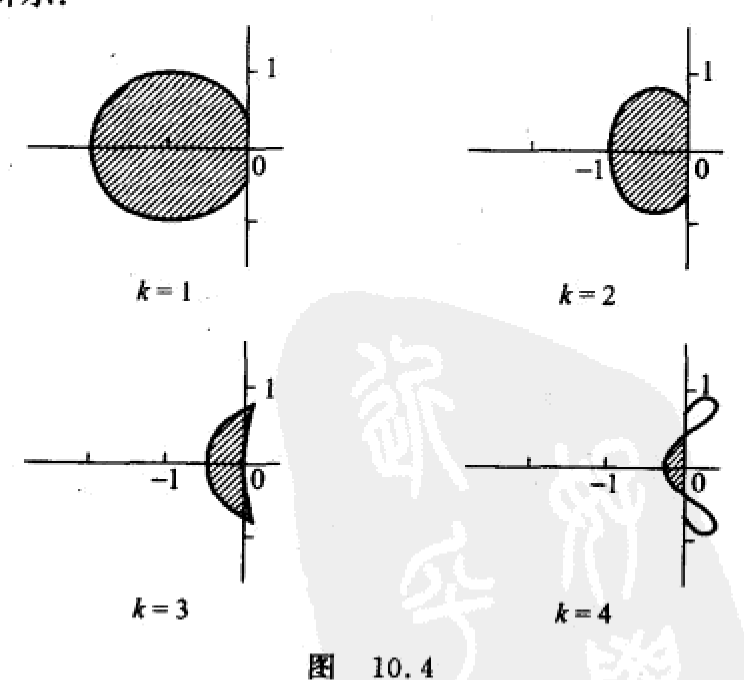


图 10.4

可见 $y(10)$的计算值和误差几乎有相同的数量级.

米尔恩方法的稳定性不好. 而汉明方法及由它构成的预测-校正方法则有较好的稳定性.

10.6 预测-校正方法

10.6.1 预测-校正的一般方法

一个隐式的线性多步法(10.1.15),其中 $\beta_k\neq0$,可以用迭代法(10.1.19)迭代求解. 迭代公式是

$$y_{n+k}^{[s+1]}+\sum_{j=0}^{k-1}\alpha_j y_{n+j}=h\beta_k f(x_{n+k},y_{n+k}^{[s]})+h\sum_{j=0}^{k-1}\beta_j f_{n+j}\quad(s=0,1,2,\cdots).\quad(10.6.1)$$

在满足定理 10.1.9 的条件下,迭代对任意初值 $y_{n+k}^{[0]}$收敛.

迭代过程(10.6.1)的每一步,均要计算一次函数值 $f(x_{n+k},y_{n+k}^{[s]})$,所以迭代计算要多次调用计算函数值 $f(x,y)$的子程序,这样要花费较大的代价. 因此可以考虑用另外一个显式方法的结果作出 y_{n+k}的估计(称为**预测**),代入隐式线性多步法(10.5.1)的右边,得到 y_{n+k} 的值(称为**校正**),这就构成**预测-校正方法**(predictor-corrector methods). 虽然迭代法(10.6.1)也是不断校正的过程,但它的迭代次数一般是由 $|y_{n+k}^{[s+1]}-y_{n+k}^{[s]}|<\varepsilon$ 来确定,而预测-校正方法则是事先规定好了预测与校正的次数.

在预测-校正方法中,除了预测和校正两步外,有时也可以加上计算函数值的步骤.

例 10.6.1 改进的欧拉方法可写成

预测: $y_{n+1}^{[0]}=y_n+hf(x_n,y_n)$.

校正：$y_{n+1}=y_n+\frac{h}{2}[f(x_n,y_n)+f(x_{n+1},y_{n+1}^{[0]})]$.

其中预测值是由欧拉公式得到，校正值由隐式的梯形公式做一次迭代得到. 改进欧拉方法也可写成

预测：$y_{n+1}^{[0]}=y_n+hf(x_n,y_n)$.

计算：$m_{n+1}=f(x'_{n+1},y_{n+1}^{[0]})$.

校正：$y_{n+1}=y_n+\frac{h}{2}[f(x_n,y_n)+m_{n+1}]$.

这里 m_{n+1} 是直接计算 $f(x_{n+1},y_{n+1}^{[0]})$ 的函数值. 有时，这种计算要经过一定的“修正”步骤，使计算精确度得到提高.

10.6.2 亚当斯预测-校正方法

分别以四阶的亚当斯显式方法和隐式方法作为预测式和校正式，可得到如下的四阶预测-校正方法.

例 10.6.2 亚当斯四阶预测-校正方法(Adams fourth-order predictor-corrector method)

预测：$y_{n+4}^{[0]}=y_{n+3}+\frac{h}{24}(55f_{n+3}-59f_{n+2}+37f_{n+1}-9f_n)$.

计算：$m_{n+4}=f(x_{n+4},y_{n+4}^{[0]})$.

校正：$y_{n+4}=y_{n+3}+\frac{h}{24}(9m_{n+4}+19f_{n+3}-5f_{n+2}+f_{n+1})$.

计算：$f_{n+4}=f(x_{n+4},y_{n+4})$.

例 10.6.3 用亚当斯四阶预测-校正方法解初值问题

$$\begin{cases}\frac{\mathrm{d}y}{\mathrm{d}x}=-y+x+1, & 0\leqslant x\leqslant 1,\\ y(0)=1.\end{cases}$$

解 取 $h=0.1$，开始值 y_1,y_2,y_3 用龙格-库塔方法计算(见例 10.4.7)，结果如下：

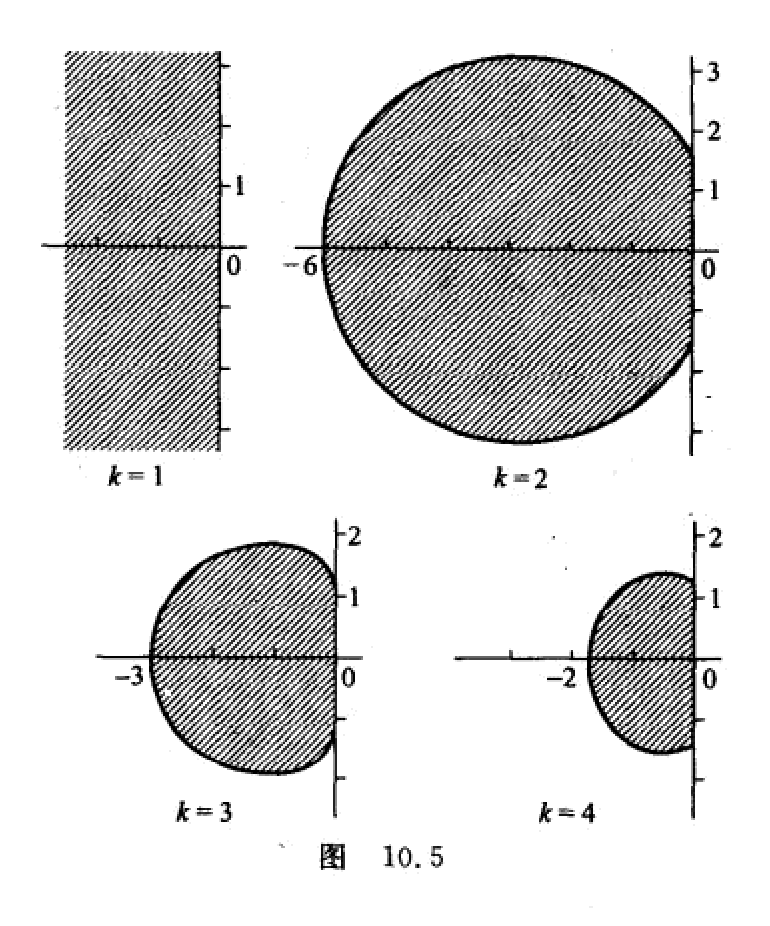


图 10.5

对于米尔恩方法,可以证明,对于任何 μ,它都不绝对稳定.

例 10.5.17 用米尔恩方法解

$$\begin{cases}\dfrac{\mathrm{d}y}{\mathrm{d}x}=-y, & 0\leqslant x\leqslant 10,\\ y(0)=1.\end{cases}$$

解 取 $h=10^{-l}(l=1,2,3,4,5)$. 计算 $y(10)$ 的近似值,结果见下表.

h	$y(10)$近似值	误　差
0.1	0.48625×10^{-4}	-0.32251×10^{-5}
0.01	0.48211×10^{-4}	-0.28114×10^{-5}
0.001	0.38639×10^{-4}	0.67607×10^{-5}
0.0001	0.32934×10^{-4}	-0.37534×10^{-4}
0.00001	0.57846×10^{-4}	-0.12446×10^{-4}

x_n	y_n	$\lvert y(x_n)-y_n \rvert$
0.4	1.07031992	1.3×10^{-7}
0.5	1.10653027	3.9×10^{-7}
0.6	1.14881103	6.1×10^{-7}
0.7	1.19658453	7.7×10^{-7}
0.8	1.24932806	9.0×10^{-7}
0.9	1.30656866	1.0×10^{-6}
1.0	1.36787837	1.1×10^{-6}

四阶亚当斯显式方法和隐式方法的主局部截断误差分别是 $\frac{251}{720}h^5y^{(5)}(x_n)$ 和 $-\frac{19}{720}h^5y^{(5)}(x_n)$. 所以对于亚当斯四阶预测-校正方法的预测值 $y_{n+4}^{[0]}$ 和校正值 y_{n+4}，有

$$y(x_{n+4})-y_{n+4}^{[0]}\approx\frac{251}{720}h^5y^{(5)}(x_n),$$

$$y(x_{n+4})-y_{n+4}\approx-\frac{19}{720}h^5y^{(5)}(x_n).$$

经过简单的运算，可得事后估计式：

$$y(x_{n+4})-y_{n+4}^{[0]}\approx-\frac{251}{270}(y_{n+4}^{[0]}-y_{n+4}),$$

$$y(x_{n+4})-y_{n+4}\approx\frac{19}{270}(y_{n+4}^{[0]}-y_{n+1}).$$

由这两个式子可对亚当斯四阶预测-校正方法进一步改进，预测值进一步取为 $y_{n+4}^{[0]}-\frac{251}{270}(y_{n+4}^{[0]}-y_{n+4})$，其中 y_{n+4} 未算出时，$y_{n+4}^{[0]}-y_{n+4}$ 用 $y_{n+3}^{[0]}-y_{n+3}$ 代替. 而校正值进一步取为 $y_{n+4}+\frac{19}{270}(y_{n+4}^{[0]}-y_{n+4})$，于是得到下面的方法.

例 10.6.4 改进的亚当斯四阶预测-校正方法

预测：$p_{n+4}=y_{n+3}+\frac{h}{24}[55f_{n+3}-59f_{n+2}+37f_{n+1}-9f_n]$.

改进：$m_{n+4}=p_{n+4}+\frac{251}{270}(c_{n+3}-p_{n+3})$，

计算：$m'_{n+4}=f(x_{n+4},m_{n+4})$.

校正：$c_{n+4}=y_{n+3}+\frac{h}{24}[9m'_{n+4}+19f_{n+3}-5f_{n+2}+f_{n+1}]$.

改进：$y_{n+4}=c_{n+4}+\frac{19}{270}(p_{n+4}-c_{n+4})$.

计算：$f_{n+4}=f(x_{n+4},y_{n+4})$.

其中的初始值 f_1,f_2,f_3 同样要另外计算，开始时可令 $c_n-p_n=0$.

10.6.3 汉明预测-校正方法

例 10.6.5 汉明预测-校正方法（Hamming predictor-corrector method）

把四阶的米尔恩方法和汉明方法相配合，并利用其主局部截断误差作出改进，可得到汉明预测-校正方法.

预测：$p_{n+4}=y_n+\frac{4h}{3}(2f_{n+1}-f_{n+2}+2f_{n+3})$.

改进：$m_{n+4}=p_{n+4}-\frac{112}{121}(p_{n+3}-c_{n+3})$.

计算：$m'_{n+4}=f(x_{n+4},m_{n+4})$.

校正：$c_{n+4}=\frac{9}{8}y_{n+3}-\frac{1}{8}y_{n+1}+\frac{3h}{8}(m'_{n+4}+2f_{n+3}-f_{n+2})$.

改进：$y_{n+4}=c_{n+4}+\frac{9}{121}(p_{n+4}-c_{n+4})$.

计算：$f_{n+4}=f(x_{n+4},y_{n+4})$.

10.7 外推方法

10.7.1 外推的一般方法

设初值问题(10.1.4),(10.1.5)的准确解是 $y(x)$，取步长为 h_i，用某种数值方法计算 N_i 步到 $x=x_0+H$. 这里 $H=N_ih_i$，记计算的近似值为 $y(x_0+H;\ h_i)$. 假设有

$$y(x_0+H;\ h_i)=y(x_0+H)+A_1h_i^{\gamma}+A_2h_i^{2\gamma}+A_3h_i^{3\gamma}+\cdots. \tag{10.7.1}$$

如果分别用两个步长 h_0,h_1 计算(其中 $h_0>h_1$),在都计算到 $x=x_0+H$ 后,分别列出式(10.7.1),消去 A_1 项,有

$$y(x_0+H;\ h_0)+\frac{y(x_0+H;\ h_0)-y(x_0+H;\ h_1)}{(h_1/h_0)^{\gamma}-1}$$
$$=y(x_0+H)-A_2h_0^{\gamma}h_1^{\gamma}+\cdots. \tag{10.7.2}$$

所以式(10.7.2)的左端就是 $y(x_0+H)$ 较好的逼近.

如果用三个步长 $h_0>h_1>h_2$ 计算,h_1 和 h_2 的计算可得到类似式(10.7.2)的式子,消去 A_2 项,便可得到更高阶的逼近.这个过程还可对 $h_0>h_1>h_2>h_3>\cdots$ 继续下去,得到

$$\begin{cases} y(x_0+H;\ h_0)=p_0^{(0)}, & & & \\ y(x_0+H;\ h_1)=p_1^{(0)}, & p_0^{(1)}, & & \\ y(x_0+H;\ h_2)=p_2^{(0)}, & p_1^{(1)}, & p_0^{(2)}, & \\ y(x_0+H;\ h_3)=p_3^{(0)}, & p_2^{(1)}, & p_1^{(2)}, & p_0^{(3)}, \\ \qquad\vdots \qquad\qquad \vdots & \vdots & \vdots & \vdots \end{cases} \tag{10.7.3}$$

式(10.7.3)表上的值可以由下式逐行确定:

$$p_i^{(j)}=p_{i+1}^{(j-1)}+\frac{p_{i+1}^{(j-1)}-p_i^{(j-1)}}{(h_i/h_{i+j})^{\gamma}-1}\quad(j=1,2,\cdots), \tag{10.7.4}$$

可以证明

$$p_i^{(j)}=y(x_0+H)+O(h_i^{\gamma}h_{i+1}^{\gamma}\cdots h_{i+j}^{\gamma}), \tag{10.7.5}$$

所以式(10.7.3)的每列比其左边的列能更快地收敛到 $y(x_0+H)$,每行也比上面的行收敛更快,而对角线比行和列收敛得更快.

使用外推方法的关键是运用表达式(10.7.1),以欧拉方法为例,可以证明,若 y_n 是计算的近似值,则有

$$y_n=y(x_n)+Ah+O(h^2).$$

例 10.7.1 用外推的欧拉方法解

$$\begin{cases} \dfrac{\mathrm{d}y}{\mathrm{d}x}=-y+x+1, & 0\leqslant x\leqslant 1, \\ y(0)=1. \end{cases}$$

$$y_0 = y(x_0),$$
$$y_1 = y_0 + h_i f(x_0, y_0),$$
$$y_{n+2} = y_n + 2h_i f(x_{n+1}, y_{n+1}) \quad (n=0,1,\cdots,N_i-1),$$
$$y(x_0+H;\ h_i) = \frac{1}{4}y_{N_i-1} + \frac{1}{2}y_{N_i} + \frac{1}{4}y_{N_i+1}.$$

利用格拉格方法计算，即可用式(10.7.3)及(10.7.4)进行外推. 其中的$\{N_i\}$可取为$\{2,4,8,16,32,\cdots\}$. 但由于每步的计算量都加倍，费时较多，所以$\{N_i\}$一般可取为$\{2,4,6,8,12,16,24,32,\cdots\}$.

例 10.7.3 用格拉格方法及式(10.7.3)，式(10.7.4)进行外推，解初值问题

$$\begin{cases}\dfrac{\mathrm{d}y}{\mathrm{d}x} = 1 - 2xy, & 0 \leqslant x \leqslant 1,\\ y(0) = 0.\end{cases}$$

解 取$H=0.1$为基本步，每基本步用格拉格方法及外推过程计算. 不外推情形实际步长为0.05，结果如表10.13. 外推情形只列出误差.

表 10.13

x_n	不外推		一次外推误差	二次外推误差
	y_n	误差		
0.0	0.00000000	0	0	0
0.1	0.09925250	8.3×10^{-5}	4.1×10^{-8}	2.0×10^{-11}
0.2	0.19458429	1.7×10^{-4}	7.1×10^{-8}	3.8×10^{-11}
0.3	0.28238289	2.5×10^{-4}	7.2×10^{-8}	5.5×10^{-11}
0.4	0.35961606	3.3×10^{-4}	3.2×10^{-8}	7.6×10^{-11}
0.5	0.42403720	4.0×10^{-4}	6.1×10^{-8}	1.1×10^{-10}
0.6	0.47430356	4.6×10^{-4}	2.1×10^{-7}	1.7×10^{-10}
0.7	0.50999998	5.0×10^{-4}	4.1×10^{-7}	2.8×10^{-10}
0.8	0.53157328	5.3×10^{-4}	6.6×10^{-7}	4.6×10^{-10}
0.9	0.54019424	5.3×10^{-4}	9.2×10^{-7}	7.1×10^{-10}
1.0	0.53757097	5.1×10^{-4}	1.2×10^{-6}	1.0×10^{-9}

解 取 $h_0=0.1$，则近似值 $y(x_n,0.1)$ 在例 10.3.1 中给出了结果. 再取 $h_1=0.05$ 进行计算，在 $x=0.1,0.2,\cdots,1$ 上得到两种方法的结果. 再外推一次，由式(10.7.4)得到

$$p_0^{(1)} = 2y(x_0+H;\ 0.05) - y(x_0+H;\ 0.1).$$

计算结果及误差如表 10.12.

表 10.12

x	$y(x;\ 0.1)$	$y(x;\ 0.05)$	$p_0^{(1)}$	误 差
0	1.000000	1.000000	1.000000	0
0.1	1.000000	1.002500	1.005000	1.63×10^{-4}
0.2	1.010000	1.014506	1.019012	2.81×10^{-4}
0.3	1.029000	1.035092	1.041184	3.66×10^{-4}
0.4	1.056100	1.063420	1.070740	4.20×10^{-4}
0.5	1.090490	1.098737	1.106984	4.53×10^{-4}
0.6	1.131441	1.140360	1.149279	4.67×10^{-4}
0.7	1.178297	1.187675	1.197053	4.68×10^{-4}
0.8	1.230467	1.240126	1.249785	4.56×10^{-4}
0.9	1.287420	1.297214	1.307008	4.38×10^{-4}
1.0	1.348678	1.358485	1.368292	4.13×10^{-4}

10.7.2 格拉格外推方法

使用外推方法时，所用数值方法应当具有式(10.7.1)的性质，而且其中 γ 应尽量大些. 利用显式的中点方法($k=2$ 的尼斯特龙方法)$y_{n+2}=y_n+2hf_{n+1}$ 可以得到 $\gamma=2$ 的(10.7.1)展开式. 格拉格(Gragg)证明了若开始值 y_1 用欧拉方法确定，而步数 N_i 总选为偶数，则用中点公式计算，总的计算方法具有 $\gamma=2$ 的(10.7.1)的展开式性质. 然而这种方法的稳定性不好. 格拉格在计算过程的最后对 $x=x_0+H$ 作了一些光滑处理，控制了不稳定性而不破坏性质(10.7.1).

例 10.7.2 格拉格外推方法(Gragg extrapolation method)

$h_i=H/N_i$，N_i 为偶数，

10.8 方程组和高阶方程的数值方法

10.8.1 一阶微分方程组的数值方法

对于一阶微分方程组初值问题(10.1.8)或其向量形式(10.1.9)，可以模拟单个方程的数值方法来构造.

例 10.8.1 一阶微分方程组的经典龙格-库塔方法

$$y_{i,n+1} = y_{in} + \frac{h}{6}[k_{1i} + 2k_{2i} + 2k_{3i} + k_{4i}] \quad (i = 1,2,\cdots,m),$$

其中

$$k_{1i} = f_i(x_n, y_{1n}, y_{2n}, \cdots, y_{mn}),$$

$$k_{2i} = f_i(x_n + \frac{h}{2}, y_{1n} + \frac{h}{2}k_{11}, y_{2n} + \frac{h}{2}k_{12}, \cdots, y_{mn} + \frac{h}{2}k_{1m}),$$

$$k_{3i} = f_i(x_n + \frac{h}{2}, y_{1n} + \frac{h}{2}k_{21}, y_{2n} + \frac{h}{2}k_{22}, \cdots, y_{mn} + \frac{h}{2}k_{2m}),$$

$$k_{4i} = f_i(x_n + h, y_{1m} + hk_{31}, y_{2n} + hk_{32}, \cdots, y_{mn} + hk_{3m}),$$

$$i = 1,2,\cdots,m.$$

求出的 y_{in} 是 $y_i(x_n)$ 的近似值.

10.8.2 高阶方程的数值方法

高阶方程的初值问题(10.1.11)化为方程组初值问题(10.1.2)后，可以按方程组的方法求解.

例 10.8.2 用经典龙格-库塔方法解初值问题

$$\begin{cases} \dfrac{d^2 y}{dx^2} - 2\dfrac{dy}{dx} + 2y = e^{2x}\sin x, & 0 \leqslant x \leqslant 1, \\ y(0) = -0.4, & \dfrac{dy(0)}{dx} = -0.6. \end{cases}$$

解 令 $y_1(x) = y(x), y_2(x) = \dfrac{dy}{dx}$，得到方程组初值问题

$$\begin{cases} \frac{dy_1}{dx} = y_2(x), & y_1(0) = -0.4, \\ \frac{dy_2}{dx} = e^{2x}\sin x - 2y_1 + 2y_2, & y_2(0) = -0.6. \end{cases}$$

按例 10.8.1 的方法计算，并与准确解 $y(x)=0.2e^{2x}(\sin x - 2\cos x)$ 比较，结果如表 10.14.

表 10.14

x_n	y_{1n}	y_{2n}	$\lvert y(x_n)-y_{1n}\rvert$
0.0	−0.40000000	−0.60000000	0
0.1	−0.46173334	−0.63163124	3.7×10^{-7}
0.2	−0.52555988	−0.64014895	8.3×10^{-7}
0.3	−0.58860144	−0.61366381	1.39×10^{-6}
0.4	−0.64661231	−0.53658203	2.03×10^{-6}
0.5	−0.69356666	−0.38873810	2.71×10^{-6}
0.6	−0.72115190	−0.14438087	3.41×10^{-6}
0.7	−0.71815295	0.22899702	4.05×10^{-6}
0.8	−0.66971133	0.77199180	4.56×10^{-6}
0.9	−0.55644290	0.15347815	4.76×10^{-6}
1.0	−0.35339886	0.25787663	4.50×10^{-6}

10.9 刚性方程组的数值方法

10.9.1 方程组的刚性现象

用一种数值方法解线性微分方程组，为保证其绝对稳定性，应选取步长 h 使得 λh 属于绝对稳定区域，其中 λ 应取遍系数矩阵的特征值. 但是在这些特征值相差十分悬殊时，计算就会有很大困难.

例 10.9.1 方程组初值问题

$$\begin{cases} \dfrac{dy_1}{dx} = -1001y_1 + 999y_2 + 2, & y_1(0) = 3, \\ \dfrac{dy_2}{dx} = 999y_1 - 1001y_2 + 2, & y_2(0) = 1. \end{cases}$$

它的系数矩阵$\begin{bmatrix} -1001 & 999 \\ 999 & -1001 \end{bmatrix}$的特征值 $\lambda_1 = -2000, \lambda_2 = -2$，方程组的解是

$$\begin{cases} y_1(x) = e^{-2000x} + e^{-2x} + 1, \\ y_2(x) = -e^{-2000x} + e^{-2x} + 1. \end{cases}$$

当 $x \to \infty$ 时，$y_1(x)$ 和 $y_2(x)$ 都趋于稳定值 1，然而它们包含的两项 e^{-2000x} 和 e^{-2x} 性质却有很大差别：前者"瞬变"过程短，约在 $x=0.01$ 就接近稳定值；后者变化慢，约在 $x=10$ 才接近稳定值.如果用经典四阶龙格-库塔方法求解，为了将"瞬变"部分表现出来，在 $0 \leqslant x \leqslant 0.01$ 应该取小步长，同时，为了达到绝对稳定，应取 $\lambda h \in (-2.78, 0)$. 由于 $\lambda_1 = -2000$，所以 $h < 0.00139$，而到达 $x = 0.01$ 后，快速瞬变部分已趋稳定，则希望取较大的步长. 但是，为了保证稳定性，仍要取小的 h 值，这样到 $x=10$ 至少要 7200 步，给计算带来很大困难，所以这类方程组要用特殊的方法求解.

定义 10.9.2 设线性微分方程组

$$\frac{d\boldsymbol{y}}{dx} = \boldsymbol{J}\boldsymbol{y} + \boldsymbol{f}(x), \tag{10.9.1}$$

常系数矩阵 $\boldsymbol{J}$ 的特征值为 $\lambda_1, \lambda_2, \cdots, \lambda_m$，若

(1) $\mathrm{Re}\lambda_j < 0 \quad (j=1,2,\cdots,m)$, (10.9.2)

(2) $s = \dfrac{\max\limits_{1 \leqslant j \leqslant m} |\mathrm{Re}\lambda_i|}{\min\limits_{1 \leqslant j \leqslant m} |\mathrm{Re}\lambda_j|} \gg 1$, (10.9.3)

则称方程组是**刚性方程组**(stiff systems)，s 称为**刚性比**(stiffness ratio).

对于一般的方程组$\dfrac{d\boldsymbol{y}}{dx} = \boldsymbol{f}(x, \boldsymbol{y})$，如果$(x, \boldsymbol{y}(x))$处的雅可比

$h\lambda$ 复平面的无限楔形区域$\{h\lambda \mid -\alpha<\pi-\arg(h\lambda)<\alpha\}$，其中 $\alpha\in\left(0,\frac{\pi}{2}\right)$，则称此方法是 $A(\alpha)$-**稳定**($A(\alpha)$-stable)的；如果对某个充分小的 $\alpha\in\left(0,\frac{\pi}{2}\right)$，方法是 $A(\alpha)$-稳定的，则称方法是 $A(0)$-**稳定的**；如果对所有 $\alpha\in\left(0,\frac{\pi}{2}\right)$，方法都是 $A(\alpha)$-稳定的，则称此方法为 $A\left(\frac{\pi}{2}\right)$-**稳定的**.

定义 10.9.7 如果一个数值方法是收敛的，且存在正常数 a，b，c，使方法在区域$\{h\lambda \mid \mathrm{Re}(h\lambda)\leqslant -a\}$上绝对稳定，且在区域$\{h\lambda \mid -a<\mathrm{Re}(h\lambda)<b, -c\leqslant \mathrm{Im}(h\lambda)\leqslant c\}$上具有高精度且相对稳定，则称此方法是**刚性稳定**(stiffly-stable)**的**.

定义 10.9.8 如果一个数值方法的绝对稳定区域包含了整个负实轴，则称该方法是 A_0-**稳定**(A_0-stable)**的**.

定义 10.9.9 如果一个 A-稳定的数值方法，用于试验方程 $y'=\lambda y$ 时有 $y_{n+1}=E(h\lambda)y_n$，当 $\mathrm{Re}(h\lambda)\to 0$ 时有 $|E(h\lambda)|\to 0$，称方法是 L-**稳定**(L-stable)或强 A-**稳定**(strongly A-stable)**的**.

各种稳定性的关系是

L- 稳定 $\Rightarrow A$- 稳定 $\Rightarrow$ 刚性稳定 $\Rightarrow A(\alpha)$- 稳定 $\Rightarrow A_0$- 稳定.

10.9.3 刚性方程组的数值方法

一种用于刚性方程组的线性多步法是下面的向后差分方法.

例 10.9.10 **吉尔方法**(Gear methods)

$$\sum_{j=1}^{k}\frac{1}{j}\nabla^{j}y_{n+k}=hf_{n+k}. \qquad (10.9.4)$$

它是隐式 k 步 k 阶的线性多步法. 写成一般线性多步法的形式

$$\sum_{j=0}^{k}\alpha_j y_{n+j}=h\beta_k f_{n+k}, \qquad (10.9.5)$$

矩阵 $J(x)=\left[\frac{\partial f(x,y)}{\partial y}\right]$ 的特征值 $\lambda_j(x)$ 对所有 $x\in I$ 满足定义 10.9.1 的(1)和(2),则称方程组在区间 I 是刚性的. $s(x)$ 称为在 x 处的刚性比.

以上关于刚性的定义不包含单个方程的情形,而且限制得过于严格. 有人提出更反映实际计算情况的定义.

定义 10.9.3 若线性微分方程组(10.9.1)满足条件:

(1) 矩阵 J 所有特征值的实部小于一个不大的正数;

(2) 矩阵 J 至少有一个特征值的实部是绝对值很大的负数;

(3) 对于具有绝对值最大负实部的特征值的解分量变化是缓慢的.

则称方程组(10.9.1)是**刚性方程组**.

10.9.2 刚性方程组的稳定性

如果一种数值方法的绝对稳定域是有限的,用它求解刚性方程组,将会导致计算步数很大. 所以绝对稳定区域有限的方法不宜用于刚性问题.

定义 10.9.4 如果一个数值方法的绝对稳定区域包含了 $\mu=h\lambda$ 复平面的整个左半平面(即 $\mathrm{Re}(h\lambda)<0$),称此方法是 *A*-**稳定**(*A*-stable)的.

定理 10.9.5 (1) 任何显式线性多步法和显式龙格-库塔方法都不可能是 *A*-稳定的.

(2) *A*-稳定的隐式线性多步法的阶不超过 2.

(3) 在 2 阶 *A*-稳定隐式线性多步法中,误差常数最小的方法是梯形方法.

定理说明 *A*-稳定的方法是很少的. 所以还要考虑其他的稳定性.

定义 10.9.6 如果一个数值方法的绝对稳定区域包含了 $\mu=$

其中 $\alpha_k=1,\alpha_0\neq0,\beta_k\neq0$. k 为 1～6 时的系数如表 10.15.

表 10.15

k	β_k	α_0	α_1	α_2	α_3	α_4	α_5	α_6
1	1	-1	1					
2	$\frac{2}{3}$	$\frac{1}{3}$	$-\frac{4}{3}$	1				
3	$\frac{6}{11}$	$-\frac{2}{11}$	$\frac{9}{11}$	$-\frac{18}{11}$	1			
4	$\frac{12}{25}$	$\frac{3}{25}$	$-\frac{16}{25}$	$\frac{36}{25}$	$-\frac{48}{25}$	1		
5	$\frac{60}{137}$	$-\frac{12}{137}$	$\frac{75}{137}$	$-\frac{200}{137}$	$\frac{300}{137}$	$-\frac{300}{137}$	1	
6	$\frac{60}{147}$	$\frac{10}{147}$	$-\frac{72}{147}$	$\frac{225}{147}$	$-\frac{400}{147}$	$\frac{450}{147}$	$-\frac{360}{147}$	1

$k=1$ 时，式(10.9.5)就是隐式欧拉方法，$k=1,2$ 时吉尔方法是A-稳定的，$k=3,4,5,6$ 时是刚性稳定和 $A(\alpha)$-稳定的，有关稳定性定义中的参数如表 10.16 所示.

表 10.16

k	1　　2	3	4	5	6
稳定性	A-稳定	$A(\alpha)$-稳定	$A(\alpha)$-稳定	$A(\alpha)$-稳定	$A(\alpha)$-稳定
α_{max}		88°27′	73°14′	51°50′	18°47′
a_{min}		0.1	0.7	2.4	6.1
c_{max}		0.75	0.75	0.75	0.5

用吉尔方法计算，每步要解 y_{n+k} 的非线性方程组，一般用牛顿法等求解，而不用类似式(10.1.19)的简单迭代法求解.

$k\geqslant7$ 时的向后差分公式(10.9.4)不满足根条件，所以实际不能使用.

另一类适用于刚性方程组的方法是隐式龙格-库塔方法. 已经证明 R 级 $2R$ 阶的隐式龙格-库塔方法是 A-稳定的，也可能是 L-稳定的. 这些方法见例 10.4.17～例 10.4.19. R 级 $2R-1$ 阶的

例 10.4.20，例 10.4.21 和 R 级 $2R-2$ 阶的例 10.4.22～例 10.4.24 也都是 A-稳定的. 用这些方法解 m 个方程的微分方程组困难在于每步要解 mR 个变量的非线性方程，用牛顿法解也要求有比较准确的初始迭代值以保证迭代收敛.

对于自守系统的方程组

$$\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}x}=\boldsymbol{f}(\boldsymbol{y}), \tag{10.9.6}$$

有一类将雅可比矩阵用于龙格-库塔法的方法. 它是隐式的，但每步只需解 R 组 m 个变量的线性方程组.

例 10.9.11 布意方法(Bui's method). 这是一个四级四阶的方法，具有 A-稳定性和 L-稳定性. 算法写成

$$\boldsymbol{y}_{n+1}=\boldsymbol{y}_n+\sum_{r=1}^{4}c_r\boldsymbol{K}_r,$$

式中 $\boldsymbol{K}_r$ 由方程

$$\left[\boldsymbol{I}-\gamma h\frac{\partial\boldsymbol{f}(\boldsymbol{y}_n)}{\partial\boldsymbol{y}}\right]\boldsymbol{K}_r=h\boldsymbol{f}\left(\boldsymbol{y}_n+\sum_{s=1}^{r-1}b_{rs}\boldsymbol{K}_s\right)\quad(r=1,2,3,4)$$

决定，其中 $\boldsymbol{I}$ 为单位阵，$\dfrac{\partial\boldsymbol{f}}{\partial\boldsymbol{y}}$为雅可比矩阵. 求解$\{\boldsymbol{K}_r\}$只有四组 m 个方程的线性方程组，而且四个方程组具有相同的系数矩阵，所以求解时每步只要做一次矩阵分解.

布意方法的系数是

$\nu=0.5728160625$,

$b_{21}=-0.5$,

$b_{31}=-0.1012236115$, $\quad b_{32}=0.9762236115$,

$b_{41}=-0.3922096763$, $\quad b_{42}=0.7151140251$,

$b_{43}=0.1430371625$,

$c_1=0.9451564786$, $\quad c_2=0.341323172$,

$c_3=0.5655139575$, $\quad c_4=-0.8519936081$.

11 常微分方程边值问题的数值方法

11.1 引言

常微分方程组**两点边值问题**(two-point boundary value problem)的一般形式是

$$\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}x}=\boldsymbol{f}(x,\boldsymbol{y}(x)),\quad a\leqslant x\leqslant b, \tag{11.1.1}$$

$$\boldsymbol{\varphi}(\boldsymbol{y}(a),\boldsymbol{y}(b))=0. \tag{11.1.2}$$

其中式(11.1.1)是一个微分方程组，$\boldsymbol{y}$ 和 $\boldsymbol{f}$ 都是有 n 个分量的向量函数. 式(11.1.2)给出在区间$[a,b]$上的边界条件，其中 $\boldsymbol{\varphi}$ 也是一个 n 维向量函数. 一般地，$n\geqslant 2$.

本章主要讨论的典型例子是二阶微分方程的两点边值问题，它可以化为 $n=2$ 时式(11.1.1)及式(11.1.2)的形式. 这种边值问题的一种形式是

$$y''=f(x,y,y'),\quad a\leqslant x\leqslant b, \tag{11.1.3}$$

$$y(a)=\alpha,\quad y(b)=\beta. \tag{11.1.4}$$

其中式(11.1.3)是$[a,b]$上一个二阶微分方程，式(11.1.4)是在$[a,b]$边界点 a 和 b 上的条件，它给出了 $x=a,b$ 上的函数值. 通常，边界条件还有其他的给法，例如线性的条件

$$\alpha_1 y(a)+\beta_1 y'(a)=\alpha,\quad \alpha_2 y(b)+\beta_2 y'(b)=\beta. \tag{11.1.5}$$

其中$|\alpha_1|+|\beta_1|\neq 0$，$|\alpha_2|+|\beta_2|\neq 0$.

在式(11.1.5)的第一式中，若 $\alpha_1\neq 0$，$\beta_1\neq 0$，称为**第三类边界条件**. 若 $\alpha_1=0$，称为**第二类边界条件**. 若 $\beta_1=0$，则化为式(11.1.4)的

形式，称为**第一类边界条件**. 式(11.1.5)的第二式也类似. 除式(11.1.5)的线性边界条件外，很多实际问题还要考虑非线性的边界条件.

边值问题(11.1.3)，(11.1.4)可能不存在解，也可能有惟一解或无穷多个解. 例如，边值问题

$$\begin{cases} y'' + \pi^2 y = 0, \\ y(0) = 0, \quad y(1) = 1 \end{cases}$$

不存在解. 而边值问题

$$\begin{cases} y'' + \pi^2 y = 0, \\ y(0) = 0, \quad y(1) = 0 \end{cases}$$

就有无穷多个解 $y(x) = c\sin\pi x$，其中 c 是任意的常数.

定理 11.1.1 设边值问题(11.1.3)，(11.1.4)中，函数 f 及 $\frac{\partial f}{\partial y}, \frac{\partial f}{\partial y'}$ 在区域

$$D = \{(x, y, y') \mid x \in [a, b], y \in \mathbb{R}, y' \in \mathbb{R}\}$$

连续，且

(1) $\frac{\partial f}{\partial y}(x, y, y') > 0, \quad \forall (x, y, y') \in D$;

(2) 存在常数 $M > 0$，使得

$$\left| \frac{\partial f}{\partial y'}(x, y, y') \right| \leqslant M, \quad \forall (x, y, y') \in D,$$

则边值问题(11.1.3)，(11.1.4)有惟一解.

定理 11.1.2 对于线性边值问题

$$y'' = p(x)y' + q(x)y + r(x), \quad a \leqslant x \leqslant b, \tag{11.1.6}$$

$$y(a) = \alpha, \quad y(b) = \beta. \tag{11.1.7}$$

若满足：

(1) p, q, r 在 $[a, b]$ 上连续；

(2) $q(x) > 0, \forall x \in [a, b]$.

则边值问题(11.1.6),(11.1.7)有惟一解.

对应于线性边值问题(11.1.6),(11.1.7),可以考虑两个初值问题:

$$y_1'' = p(x)y_1' + q(x)y_1 + r(x), \quad a \leqslant x \leqslant b, \tag{11.1.8}$$

$$y_1(a) = \alpha, \quad y_1'(a) = 0, \tag{11.1.9}$$

以及

$$y_2'' = p(x)y_2' + q(x)y_2, \quad a \leqslant x \leqslant b, \tag{11.1.10}$$

$$y_2(a) = 0, \quad y_2'(a) = 1. \tag{11.1.11}$$

定理 11.1.3 设 y_1 是初值问题(11.1.8),(11.1.9)的解,y_2 是初值问题(11.1.10),(11.1.11)的解,并设 $y_2(b) \neq 0$,则边值问题(11.1.6),(11.1.7)的解为

$$y(x) = y_1(x) + \frac{\beta - y_1(b)}{y_2(b)} y_2(x). \tag{11.1.12}$$

11.2 打靶法

11.2.1 线性边值问题的打靶法

解两点边值问题的**打靶法**(shooting method)是通过解初值问题的途径求解边值问题的一类方法. 对于线性的边值问题(11.1.6),(11.1.7),用初值问题的数值方法(例如龙格-库塔方法,亚当斯方法等)分别解初值问题(11.1.8),(11.1.9)和初值问题(11.1.10),(11.1.11),得到 $y_1(x)$ 和 $y_2(x)$ 的数值解,然后利用定理 11.1.3 的公式(11.1.12)得到边值问题(11.1.6),(11.1.7)的数值解,这就是线性边值问题的打靶法.

定理 11.2.1 在以上描述的打靶法中,若 y_{1n} 及 y_{2n} 分别为 $y_1(x)$ 和 $y_2(x)$ 在节点 $x_n(n=0,1,\cdots,N)$ 的函数值的 $O(h^p)$ 近似值,则按公式(11.1.12)算得的结果 y_n 亦为 $y(x_n)$ 的 $O(h^p)$ 近似值,且存在常数 $K>0$,使得

$$|y(x_n)-y_n|\leqslant Kh^p\left|1+\frac{y_{2n}}{y_{1n}}\right|. \qquad (11.2.1)$$

例 11.2.2 用打靶法解线性边值问题

$$\begin{cases} y''=-\dfrac{2}{x}y'+\dfrac{2}{x^2}y+\dfrac{\sin(\ln x)}{x^2}, & 1\leqslant x\leqslant 2, \\ y(1)=1, \quad y(2)=2. \end{cases}$$

解 根据定理 11.1.3,本例求解边值问题可以化为解两个初值问题

$$\begin{cases} y''_1=-\dfrac{2}{x}y'_1+\dfrac{2}{x^2}y_1+\dfrac{\sin(\ln x)}{x^2}, & 1\leqslant x\leqslant 2, \\ y_1(1)=1, \quad y'_1(1)=0. \end{cases}$$

$$\begin{cases} y''_2=-\dfrac{2}{x}y'_2+\dfrac{2}{x^2}y_2, & 1\leqslant x\leqslant 2, \\ y(1)=0, \quad y'_2(1)=1. \end{cases}$$

分别用经典龙格-库塔方法求解. 取 $h=0.1$,得到 y_{1n}, y_{2n}. 再由 $y_{1n}+\dfrac{2-y_{1N}}{y_{2N}}y_{2n}$ 计算 y_n,计算结果如表 11.1.

表 11.1

x_n	y_{1n}	y_{2n}	y_n	准确解 $y(x_n)$	$\lvert y(x_n)-y_n\rvert$
1.0	1.00000000	0.00000000	1.00000000	1.00000000	—
1.1	1.00896058	0.09117986	1.09262917	1.09262930	1.43×10^{-7}
1.2	1.03245472	0.16851175	1.18708471	1.18708484	1.34×10^{-7}
1.3	1.06674375	0.23608704	1.28338227	1.28338236	9.78×10^{-8}
1.4	1.10928795	0.29659067	1.38144589	1.38144595	6.02×10^{-8}
1.5	1.15830000	0.35184379	1.48115939	1.48115942	3.06×10^{-8}
1.6	1.21248372	0.40311695	1.58239245	1.58239246	1.08×10^{-8}
1.7	1.27087454	0.45131840	1.68501396	1.68501396	5.43×10^{-10}
1.8	1.33273851	0.49711137	1.78889854	1.78889853	5.05×10^{-9}
1.9	1.39750618	0.54098928	1.89392951	1.89392951	4.41×10^{-9}
2.0	1.46472815	0.58332538	2.00000000	2.00000000	—

例 11.2.2 用打靶法计算有比较好的效果，但在别的例子中，有时由于误差的作用，用式(11.1.12)计算会引起有效数字的消失，因而得不到满意的结果. 有时可以用相反方向的打靶法解边值问题，即分别解初值问题

$$\begin{cases} y_1'' = p(x)y_1' + q(x)y_1 + r(x), \quad a \leqslant x \leqslant b, \\ y_1(b) = \beta, \quad y_1'(b) = 0; \end{cases}$$

$$\begin{cases} y_2'' = p(x)y_2' + q(x)y_2, \\ y_2(b) = 0, \quad y_2'(b) = 1. \end{cases}$$

设它们的解分别是 $y_1(x)$ 和 $y_2(x)$，类似式(11.1.12)有

$$y(x) = y_1(x) + \frac{\alpha - y_1(a)}{y_2(a)} y_2(x). \tag{11.2.2}$$

11.2.2 非线性边值问题的打靶法

在二阶方程边值问题(11.1.3)，(11.1.4)是非线性的情况下，对参数 t，初值问题

$$y'' = f(x, y, y'), \quad a \leqslant x \leqslant b, \tag{11.2.3}$$

$$y(a) = \alpha, \quad y'(a) = t \tag{11.2.4}$$

的解记为 $y(x,t)$. 如果此解与边值问题(11.1.3)，(11.1.4)的解相同，即

$$y(b,t) = \beta \tag{11.2.5}$$

则可用解初值问题代替解边值问题. 但是参数 t 并不是很容易确定的，所以用序列 $\{t_k\}$ 来逼近它.

定理 11.2.3 设在区域

$$D = \{(x,y,y') \mid a \leqslant x \leqslant b, -\infty < y < +\infty, -\infty < y' < +\infty\}$$

上，$f(x,y,y')$ 满足：

(1) $f, \dfrac{\partial f}{\partial y}, \dfrac{\partial f}{\partial y'}$ 连续；

(2) 存在常数 $M>0$，使 $\left|\dfrac{\partial f}{\partial y'}\right| \leqslant M$；

(3) 存在常数 $L>0$,使 $0<\frac{\partial f}{\partial y}<L$.

则边值问题(11.1.3),(11.1.4)及初值问题(11.2.3),(11.2.4)对于任意参数 t,均在$[a,b]$上存在惟一的解.

解边值问题(11.1.3),(11.1.4)的**打靶法**是用数值方法解初值问题

$$y'' = f(y,y,y'), \quad a \leqslant x \leqslant b, \tag{11.2.6}$$

$$y(a) = \alpha, \quad y'(a) = t_k, \tag{11.2.7}$$

得到解 $y(x,t_k)$. 若对给定的误差限 ε,有 $|y(b,t_k)-\beta|<\varepsilon$,则认为 $y(x,t_k)$是边值问题(11.1.3),(11.1.4)的解,否则,按一定的方法将 t_k 修改为 t_{k+1},重复上述计算.

要想成功地运用打靶法,就要求$\lim\limits_{k\to\infty}y(b,t_k)=\beta$. 实际上,$\{t_k\}$可以看成非线性方程(11.2.5)的一个近似解序列,所以打靶法的具体计算步骤,可以采用解非线性方程的一些方法(参见 5.2 节).

例 11.2.4　二分法(bisection method)

记

$$F(t_k) = y(b,t_k) - \beta. \tag{11.2.8}$$

在打靶法过程开头,选初值 t_1 和 t_2,使 $F(t_1)F(t_2)<0$. 一般地,若已知 $t_k<t_{k+1}$,可以构成迭代区间$[t_k,t_{k+1}]$. 设 $F(t_k)F(t_{k+1})<0$,令

$$t_{k+2} = \frac{1}{2}(t_k + t_{k+1}),$$

将$[t_k, t_{k+1}]$分半,若 $F(t_{k+2})F(t_k)<0$,则以$[t_k, t_{k+2}]$代替$[t_k,t_{k+1}]$,否则以$[t_{k+2},t_{k+1}]$代替$[t_k,t_{k+1}]$,继续进行迭代.

例 11.2.5　割线法(secant method)

在打靶法中,选定初值 t_1,t_2. 一般地,若已知 t_k,t_{k+1},则令

$$t_{k+2} = t_{k+1} - \frac{[y(b,t_{k+1})-\beta](t_{k+1}-t_k)}{y(b,t_{k+1})-y(b,t_k)}. \tag{11.2.9}$$

牛顿法是解非线性方程的一种有效方法. 把它用于打靶法,即

要解方程(11.2.5),要用到$\frac{\partial y(b,t)}{\partial t}$的值,而 $y(b,t)$的表达式是未知的,如果把 $y(x,t)$看成初值问题(11.2.3),(11.2.4)的解,两式分别对 t 求导,有

$$\frac{\partial y''}{\partial t}=\frac{\partial f}{\partial y}\frac{\partial y}{\partial t}+\frac{\partial f}{\partial y'}\frac{\partial y'}{\partial t},$$

$$\frac{\partial y(a,t)}{\partial t}=0,\quad \frac{\partial y'(a,t)}{\partial t}=1.$$

以 $z(x,t)$记$\frac{\partial y(x,t)}{\partial t}$,并设对 x 和 t 的导数可交换次序,于是可以得到关于 z 的一个微分方程的初值问题

$$z''=\frac{\partial f(x,y,y')}{\partial y}z+\frac{\partial f(x,y,y')}{\partial y'}z',\quad a\leqslant x\leqslant b,\tag{11.2.10}$$

$$z(a)=0,\quad z'(a)=1.\tag{11.2.11}$$

所以牛顿法用于打靶法描述如下.

例 11.2.6 (牛顿法)在打靶法中,给定初值 t_1.一般地,若已知 t_k,从初值问题(11.2.6),(11.2.7)求解出 $y(x,t_k)$,再从初值问题(11.2.10),(11.2.11)求解出 $z(x,t)$,则

$$t_{k+1}=t_k-\frac{y(b,t_k)-\beta}{z(b,t_k)}.\tag{11.2.12}$$

例 11.2.7 用打靶法的牛顿迭代过程解边值问题

$$\begin{cases}y''=\frac{1}{8}(32+2x^3-yy'),\quad 1\leqslant x\leqslant 3,\\ y(1)=17,\quad y(3)=\frac{43}{3}.\end{cases}$$

解 由例 11.2.6 的描述,把边值问题化成每步求解两个初值问题:

$$\begin{cases}y''=\frac{1}{8}(32+2x^3-yy'),\quad 1\leqslant x\leqslant 3,\\ y(1)=17,\quad y'(1)=t_k.\end{cases}$$

$$
\begin{cases} z'' = -\dfrac{1}{8}(yz' + y'z), & 1 \leqslant x \leqslant 3, \\ z(1) = 0, & z'(1) = 1. \end{cases}
$$

给定初值 t_1，根据式(11.2.12)进行迭代，直到

$$
\left| y(b, t_k) - \frac{43}{3} \right| < \varepsilon
$$

时迭代结束.

11.3 有限差分方法

11.3.1 线性边值问题的差分方法

二阶线性方程的第一类两点边值问题是

$$
y'' = p(x)y' + q(x)y + r(x), \quad a \leqslant x \leqslant b, \tag{11.3.1}
$$

$$
y(a) = \alpha, \quad y(b) = \beta. \tag{11.3.2}
$$

其中式(11.3.2)为第一类边界条件，也可以考虑第二和第三类边界条件：

$$
y'(a) = \alpha, \quad y'(b) = \beta, \tag{11.3.3}
$$

$$
y'(a) - \alpha_1 y(a) = \alpha, \quad y'(b) + \beta_1 y(b) = \beta, \tag{11.3.4}
$$

其中 $\alpha_1 \geqslant 0, \beta_1 \geqslant 0$. 对于左端点 $x = a$ 和右端点 $x = b$，也可以分别取不同类型的边界条件.

在方程(11.3.1)的两边同乘 $e^{\int p(x)dx}$，再令 $t = \int e^{\int p(x)dx} dx$，方程化为

$$
\frac{d^2 y}{dt^2} + Q(t)y = R(t).
$$

所以有时考虑方程(11.3.1)中 $p(x) \equiv 0$.

用节点 $x_i (i = 0, 1, \cdots, N)$ 将区间$[a, b]$ N 等分，记 $x_i = a + ih$ $(i = 0, 1, \cdots, N)$，$h = \dfrac{b - a}{N}$. 设 $y \in C^4[a, b]$，由泰勒公式有

$$y(x_{i+1}) = y(x_i) + hy'(x_i) + \frac{h^2}{2}y''(x_i) + \frac{h^3}{6}y'''(x_i) + \frac{h^4}{24}y^{(4)}(\xi_i^+),$$

$$y(x_{i-1}) = y(x_i) - hy'(x_i) + \frac{h^2}{2}y''(x_i) - \frac{h^3}{6}y'''(x_i) + \frac{h^4}{24}y^{(4)}(\xi_i^-),$$

其中 $\xi_i^+ \in (x_i, x_{i+1})$，$\xi_i^- \in (x_{i-1}, x_i)$. 对此两式用中值定理得到

$$y''(x_i) = \frac{1}{h^2}[y(x_{i+1}) - 2y(x_i) + y(x_{i-1})] - \frac{h^2}{12}y^{(4)}(\xi_i), \tag{11.3.5}$$

其中 $\xi_i \in (x_{i-1}, x_{i+1})(i=1,2,\cdots,n-1)$.

同理可得

$$y'(x_i) = \frac{1}{2h}[y(x_{i+1}) - y(x_{i-1})] - \frac{h^2}{6}y'''(\eta_i), \tag{11.3.6}$$

其中 $\eta_i \in (x_{i-1}, x_{i+1})(i=1,2,\cdots,n-1)$. 式(11.3.6)称为 $y'(x_i)$ 的**中心差分公式**.

将式(11.3.5)，式(11.3.6)代入式(11.3.1)，得到

$$\frac{y(x_{i+1}) - 2y(x_i) + y(x_{i-1})}{h^2} = p(x_i)\left[\frac{y(x_{i+1}) - y(x_{i-1})}{2h}\right] + q(x_i)y(x_i) + r(x_i) - \frac{h^2}{12}[2p(x_i)y'''(\eta_i) - y^{(4)}(\xi_i)].$$

将上面方程最后一项略去，加上边界条件(11.3.2)，就得到解边值问题(11.3.1)，(11.3.2)截断误差为 $O(h^2)$ 的差分方程组：

$$\begin{cases} y_0 = \alpha, \\ \dfrac{y_{i+1} - 2y_i + y_{i-1}}{h^2} - p(x_i)\dfrac{y_{i+1} - y_{i-1}}{2h} - q(x_i)y_i = r(x_i) \\ \qquad\qquad\qquad\qquad (i = 1,2,\cdots,N-1) \\ y_N = \beta, \end{cases} \tag{11.3.7}$$

其中 $y_i(i=0,1,\cdots,n)$ 是 $y(x_i)$ 的近似值，方程组写成矩阵形式为

$$Ay = b, \tag{11.3.8}$$

其中

$$y = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{N-1} \\ y_N \end{pmatrix}, \quad b = \begin{pmatrix} \alpha \\ h^2 r(x_1) \\ \vdots \\ h^2 r(x_{N-1}) \\ \beta \end{pmatrix},$$

$$A = \begin{pmatrix} 1 & 0 & & & & \\ 1+\frac{h}{2}p(x_1) & -(2+h^2q(x_1)) & 1-\frac{h}{2}p(x_1) & & & \\ & 1+\frac{h}{2}p(x_2) & -(2+h^2q(x_2)) & 1-\frac{h}{2}p(x_2) & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1+\frac{h}{2}p(x_{N-1}) & -(2+h^2q(x_{N-1})) & 1-\frac{h}{2}p(x_{N-1}) \\ & & & & 0 & 1 \end{pmatrix}.$$

差分方程组(11.3.7)是一个三对角的线性代数方程组，可以用追赶法来解. 若解出了 $y=(y_0, y_1, \cdots, y_N)^{\mathrm{T}}$，就求得 $y(x_i)$ $(i=0,1,\cdots,N)$的近似值.

对于第二、三类边界条件(11.3.3)和(11.3.4)，导数值也要用差分近似，例如可以用公式

$$y'(x_0) = \frac{y(x_1)-y(x_0)}{h} + O(h),$$

$$y'(x_N) = \frac{y(x_N)-y(x_{N-1})}{h} + O(h).$$

为了使截断误差也达到 $O(h^2)$，可用牛顿等距节点插值公式

$$y'(x_0) = \frac{-y(x_2)+4y(x_1)-3y(x_0)}{2h} + O(h^2),$$

$$y'(x_N) = \frac{3y(x_N)-4y(x_{N-1})+y(x_{N-2})}{2h} + O(h^2).$$

这样，对第三类边值问题(11.3.1)，(11.3.4)，得到差分方程组

$$\begin{cases}\dfrac{-y_2+4y_1-3y_0}{2h}-\alpha_1 y_0=\alpha, \\ \dfrac{y_{i+1}-2y_i+y_{i-1}}{h^2}-p(x_i)\dfrac{y_{i+1}-y_{i-1}}{2h}-q(x_i)y_i=r(x_i), \\ \qquad\qquad\qquad (i=1,2,\cdots,N-1) \\ \dfrac{3y_N-4y_{N-1}+y_{N-2}}{2h}+\beta_1 y_N=\beta. \end{cases} \tag{11.3.9}$$

类似地可得出第二类边值问题的差分方程组.

定理 11.3.1 设在$[a,b]$上 $q(x)\geqslant 0$，$L=\max\limits_{a\leqslant x\leqslant b}|p(x)|$，则当 $h<\dfrac{2}{L}$时，方程组(11.3.7)存在惟一的解.

定理 11.3.2 设在$[a,b]$上 $q(x)\geqslant 0$，$p(x)\equiv 0$，边值问题(11.3.1)，(11.3.2)的解 $y\in C^4[a,b]$，$M_4=\max\limits_{a\leqslant x\leqslant b}|y^{(4)}(x)|$，差分方程组(11.3.7)的解为 $y_i(i=0,1,\cdots,N)$，则有

$$|y(x_i)-y_i|<\frac{M_4(b-a)^2}{96}h^2 \quad (i=0,1,\cdots,N). \tag{11.3.10}$$

定理 11.3.2 说明，若 $x=x_i$ 固定，当 $h\to 0$ 时，差分方程的解收敛到微分方程边值问题的解.

例 11.3.3 用差分方法解线性边值问题

$$\begin{cases} y''=-\dfrac{2}{x}y'+\dfrac{2}{x^2}y+\dfrac{\sin(\ln x)}{x^2}, \quad 1\leqslant x\leqslant 2, \\ y(1)=1, \quad y(2)=2. \end{cases}$$

解 设 $h=0.1$，$N=10$. 用上述方法列出差分方程(11.3.7)，用追赶法求解，结果如表 11.2.

表 11.2

x_i	y_i	$y(x_i)$	$\lvert y_i-y(x_i)\rvert$
1.0	1.00000000	1.00000000	—
1.1	1.09260052	1.09262930	2.88×10^{-5}
1.2	1.18704313	1.18708484	4.17×10^{-5}
1.3	1.28333687	1.28338236	4.55×10^{-5}
1.4	1.38140205	1.38144595	4.39×10^{-5}
1.5	1.48112026	1.48115942	3.92×10^{-5}
1.6	1.58235990	1.58239246	3.26×10^{-5}
1.7	1.68498902	1.68501396	2.49×10^{-5}
1.8	1.78888175	1.78889853	1.68×10^{-5}
1.9	1.89392110	1.89392951	8.41×10^{-6}
2.0	2.00000000	2.00000000	—

与例 11.2.2 比较，这里用差分方法的精确度不如例 11.2.2，因为在例 11.2.2 中用的是四阶龙格-库塔方法，而本例使用的差分方法的截断误差是 $O(h^2)$. 虽然可以构造截断误差更高阶的差分方程，例如截断误差是 $O(h^4)$，但由于这样近似 $y''(x_i)$ 时要用到 $y(x_{i-2})$ 和 $y(x_{i+2})$ 的值，使差分方程组更为复杂，故一般较少用.

可以运用类似 10.7 节的外推方法，或类似第 4 章中的龙格积分算法，对差分方法的结果进行外推.

例 11.3.4 用差分方法的外推过程解例 11.3.3 的边值问题.

解 设 $h=0.1, 0.05$ 及 0.025，作第一次外推为

$$\mathrm{Ext}_{1i}=\frac{1}{3}[4y_i(h=0.05)-y_i(h=0.1)],$$

第二次和第三次外推为

$$\mathrm{Ext}_{2i}=\frac{1}{3}[4y_i(h=0.025)-y_i(h=0.05)],$$

$$\mathrm{Ext}_{3i} = \frac{1}{15}[16\mathrm{Ext}_{2i} - \mathrm{Ext}_{1i}].$$

计算结果如表 11.3，其中 Ext_{3i}列出的是小数位经过修正的结果，事实上计算结果与准确解在节点上的最大误差为 6.3×10^{-11}，外推的修正效果是很好的. 表 11.3 中略去了 $y_i(h=0.1)$之值，此栏参见例 11.3.3.

表 11.3

x_i	$y_i(h=0.05)$	$y_i(h=0.025)$	Ext_{1i}	Ext_{2i}	Ext_{3i}
1.0	1.00000000	1.00000000	1.00000000	1.00000000	1.00000000
1.1	1.09262207	1.09262479	1.09262925	1.09262930	1.09262930
1.2	1.18707436	1.18708222	1.18708477	1.18708484	1.18708484
1.3	1.28337094	1.28337950	1.28338230	1.28338236	1.28338236
1.4	1.38143493	1.38144319	1.38144589	1.38144595	1.38144595
1.5	1.48114959	1.48115696	1.48115937	1.48115941	1.48115942
1.6	1.58238429	1.58239042	1.58239242	1.58239246	1.58239246
1.7	1.68500770	1.68501240	1.68501393	1.68501396	1.68501396
1.8	1.78889432	1.78889748	1.78889852	1.78889853	1.78889853
1.9	1.89392740	1.89392898	1.89392950	1.89392951	1.89392951
2.0	2.00000000	2.00000000	2.00000000	2.00000000	2.00000000

11.3.2 非线性边值问题的差分方法

对于一般的二阶非线性方程边值问题(11.1.3)，(11.1.4)，可类似地建立差分方程. 为保证边值问题存在惟一解，应设函数 f 满足定理 11.1.1 的条件，并设

$$L = \max_{(x,y,y')\in D}\left|\frac{\partial f}{\partial y}(x,y,y')\right|, \qquad (11.3.11)$$

其中区域 D 如定理 11.1.1 所作的规定.

将式(11.3.5)和式(11.3.6)用于方程(11.1.3)，并略去误差，

加上边界条件，就得到边值问题的解在节点上的近似值 $y_0,y_1,\cdots,y_N$ 满足的非线性方程组：

$$\begin{cases} y_0 = \alpha, \\ \dfrac{y_{i+1}-2y_i+y_{i-1}}{h^2} - f\left(x_i, y_i, \dfrac{y_{i+1}-y_{i-1}}{2h}\right) = 0 \\ \qquad\qquad\qquad\qquad (i = 1,2,\cdots,N-1), \\ y_N = \beta. \end{cases} \tag{11.3.12}$$

定理 11.3.5 设边值问题(11.1.3),(11.1.4)满足定理 11.1.1 的条件，如果 $h \leqslant \dfrac{2}{L}$，则方程组(11.3.12)有惟一的解，其中 L 如式(11.3.11)所示.

解方程组(11.3.12)，可以选用解非线性方程组的牛顿法.

将方程组(11.3.12)中的 $y_0=\alpha, y_N=\beta$ 代入 $i=1$ 和 $i=N-1$ 的方程中，未知数写成向量 $\boldsymbol{y}=(y_1,y_2,\cdots,y_{N-1})^{\mathrm{T}}$，则牛顿法的迭代公式为

$$\boldsymbol{y}^{(k)} = \boldsymbol{y}^{(k-1)} - \boldsymbol{J}(\boldsymbol{y}^{(k-1)})^{-1}\boldsymbol{F}(\boldsymbol{y}^{(k-1)}), \tag{11.3.13}$$

其中 $\boldsymbol{J}(\boldsymbol{y})$ 是方程组左端函数的雅可比矩阵，可写成一个三对角矩阵，第 i 行第 j 列的元素是

$$\boldsymbol{J}(\boldsymbol{y})_{ij} = \begin{cases} -1+\dfrac{h}{2}f_{y'}\left(x_i, y_i, \dfrac{y_{i+1}-y_{i-1}}{2h}\right), & i=j-1, j=2,3,\cdots,N-1, \\ 2+h^2 f_y\left(x_i, y_i, \dfrac{y_{i+1}-y_{i-1}}{2h}\right), & i=j, j=1,2,\cdots,N-1, \\ -1-\dfrac{h}{2}f_{y'}\left(x_i, y_i, \dfrac{y_{i+1}-y_{i-1}}{2h}\right), & i=j+1, j=1,2,\cdots,N-2. \end{cases}$$

其中 $y_0=\alpha, y_N=\beta$.

每迭代一步要解的三对角方程组是

$$
\boldsymbol{J}(\boldsymbol{y})\begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_{N-2} \\ v_{N-1} \end{pmatrix} = -\begin{pmatrix} 2y_1 - y_2 - \alpha + h^2 f\left(x_1, y_1, \dfrac{y_2 - \alpha}{2h}\right) \\ -y_1 + 2y_2 - y_3 + h^2 f\left(x_2, y_2, \dfrac{y_3 - y_1}{2h}\right) \\ \vdots \\ -y_{N-3} + 2y_{N-2} - y_{N-1} + h^2 f\left(x_{N-2}, y_{N-2}, \dfrac{y_{N-1} - y_{N-3}}{2h}\right) \\ -y_{N-2} + 2y_{N-1} - \beta + h^2 f\left(x_{N-1}, y_{N-1}, \dfrac{\beta - y_{N-2}}{2h}\right) \end{pmatrix},
$$

其中系数矩阵和右端向量中的 y_i 均取 $y_i^{(k-1)}$. 解出 v_i 后，令 $y_i^{(k)} = y_i^{(k-1)} + v_i (i=1,2,\cdots,N-1)$，即完成一次迭代.

11.4 变分方法

自伴随的边值问题是在实际问题中一类十分重要的二阶线性方程的边值问题. 自伴随的二阶线性方程是

$$
-\frac{\mathrm{d}}{\mathrm{d}x}\left(p(x)\frac{\mathrm{d}y}{\mathrm{d}x}\right) + q(x)y = f(x) \quad (a \leqslant x \leqslant b), \tag{11.4.1}
$$

其中

$$
p \in C^1[a,b], \quad p(x) \geqslant p_0 > 0,
$$
$$
q \in C[a,b], \quad q(x) \geqslant 0.
$$

在 $x=a$ 和 $x=b$ 的边界条件，可以分别取为第一、二、三类边界条件，如式(11.1.4)和式(11.1.5)所示.

方程(11.4.1)的物理背景可视为弹性杆的平衡问题，其中 $y=y(x)$ 为杆的纵向位移. 在第一类边界条件的情形，杆的总能量为

$$
I(y) = \frac{1}{2}\int_a^b \left[p(x)\left(\frac{\mathrm{d}y}{\mathrm{d}x}\right)^2 + q(x)(y(x))^2 - 2f(x)y(x)\right]\mathrm{d}x. \tag{11.4.2}
$$

如果给定了函数 y，$I(y)$就确定了一个对应的实数值，所以称 I 为一个"泛函".

11.4.1 变分问题

对于方程(11.4.1)的第一边值问题

$$-\frac{\mathrm{d}}{\mathrm{d}x}\left(p(x)\frac{\mathrm{d}y}{\mathrm{d}x}\right)+q(x)y=f(x)\quad(a\leqslant x\leqslant b),\tag{11.4.3}$$

$$y(a)=y_a,\quad y(b)=y_b.\tag{11.4.4}$$

可以提出与之相关的泛函极小值问题，即**里茨变分问题**(Ritz's variational problem)：

求 $y\in K$，使得对一切 $w\in K$，有

$$I(y)\leqslant I(w).\tag{11.4.5}$$

其中 $I(y)$如式(11.4.2)所示，而

$$K=\{y\mid y\in C^1[a,b],y(a)=y_a,y(b)=y_b\}.\tag{11.4.6}$$

另外一种形式的变分问题是**伽辽金变分问题**(Galerkin variational problem)：

求 $y\in K$，使得

$$D(y,w)-F(w)=0\tag{11.4.7}$$

对一切 $w\in K_0$ 成立，其中

$$D(y,w)=\int_a^b\left[p(x)\frac{\mathrm{d}y}{\mathrm{d}x}\frac{\mathrm{d}w}{\mathrm{d}x}\mathrm{d}x+q(x)yw\right]\mathrm{d}x,\tag{11.4.8}$$

$$F(w)=\int_a^b f(x)w\mathrm{d}x,\tag{11.4.9}$$

K 如式(11.4.6)所示，而

$$K_0=C_0^1[a,b]=\{y\mid y\in C^1[a,b],y(a)=0,y(b)=0\}.\tag{11.4.10}$$

如果边值问题(11.4.1)，(11.4.3)的物理背景理解为两端固定的杆的平衡问题，则里茨变分问题可以理解为对应的最小

势能原理,伽辽金变分问题则可理解为**虚功原理**.三者有下面的关系.

定理 11.4.1 如果函数 y 满足边值问题(11.4.1),(11.4.3),则 y 是伽辽金变分问题(11.4.7)的解.反之,如果 y 是伽辽金变分问题(11.4.7)的解,且 $y\in C^1[a,b]\cap C^2(a,b)$,则 y 满足边值问题(11.4.1).如果 y 是里茨变分问题(11.4.5)的解,则 y 是伽辽金变分问题(11.4.7)的解,反之亦然.

为了简化问题,设边界条件(11.4.3)中 $y_a=y_b=0$.这样,在变分问题中,$K=K_0$.如果 $y(x)$ 满足非齐次的边界条件,即 $y_a\neq 0$ 或 $y_b\neq 0$,则令

$$\tilde{y}(x)=y(x)-y_a-\frac{y_b-y_a}{b-a}(x-a),$$

可验证 $\tilde{y}(a)=\tilde{y}(b)=0$,即 $\tilde{y}(x)$ 满足齐次边界条件.

定理 11.4.1 指出了里茨变分问题与伽辽金变分问题的等价性,但这仅对自伴随的边值问题成立.方程(11.4.1)中 $p(x)\geqslant p_0>0$, $q(x)\geqslant 0$,满足 $D(y,w)=D(w,y)$(对一切 $y,w\in K$),且对任意的 $y\in C^1[a,b]$, $D(y,y)\geqslant 0$.至于一些非自伴随的边值问题,仍可有伽辽金变分问题,所以它比里茨变分问题使用更为广泛.

用差分方法求解边值问题(11.4.1),(11.4.3)时,要处理 y 的二阶导数,而求解里茨变分问题或伽辽金变分问题时,只要处理 y 的一阶导数.常常把问题(11.4.7)称为边值问题(11.4.1),(11.4.3)的弱形式.

对于第二或第三类边值问题,也可提出类似的变分问题.

例 11.4.2 如果边值问题中微分方程仍为(11.4.1),边界条件为

$$y(a)=0,\quad \frac{\mathrm{d}y(b)}{\mathrm{d}x}=0.$$

则仍有类似的里茨变分问题和伽辽金变分问题，其中 $I(y)$ 仍为式(11.4.2)所示，而 $D(y,w)$ 和 $F(w)$ 仍如(11.4.8)和(11.4.9)所示，但是

$$K = K_0 = \{y \mid y \in C^1[a,b], y(a) = 0\}.$$

在例 11.4.2 的变分问题中，并没有要求函数集合 K 中的函数满足第二类边界条件 $\frac{\mathrm{d}y(b)}{\mathrm{d}x}=0$，因而对变分问题的解 y 也没有提此要求. 但是，类似定理 11.4.1，如果 y 是变分问题的解，且 $y\in C^1[a,b]\cap C^2(a,b)$，则 y 满足边值问题，所以变分问题的解是自然满足第二类边界条件的. 称例 11.4.2 的边界条件 $\frac{\mathrm{d}y(b)}{\mathrm{d}x}=0$ 为**自然边界条件**(natural boundary condition). 而第一类边界条件 $y(a)=0$ 称为**本质边界条件**(essential boundary condition)或**强加边界条件**(forced boundary condition).

例 11.4.3 若边值问题为方程(11.4.1)及边界条件

$$\left(-p(x)\frac{\mathrm{d}y}{\mathrm{d}x}+\alpha_1 y(x)\right)\Big|_{x=a} = g_1,$$

$$\left(p(x)\frac{\mathrm{d}y}{\mathrm{d}x}+\alpha_2 y(x)\right)\Big|_{x=b} = g_2,$$

其中 $\alpha_1\geqslant 0,\alpha_2\geqslant 0$，则里茨变分问题和伽辽金变分问题形式仍如式(11.4.5)和式(11.4.7)所示，其中

$$D(y,w) = \int_a^b\left[p(x)\frac{\mathrm{d}y}{\mathrm{d}x}\frac{\mathrm{d}w}{\mathrm{d}x}+q(x)yw\right]\mathrm{d}x + \alpha_1 y(a)w(a)+\alpha_2 y(b)w(b),$$

$$F(w) = \int_a^b f(x)w\,\mathrm{d}x + g_1 w(a) + g_2 w(b),$$

$$I(y) = \frac{1}{2}D(y,y) - F(y),$$

$$K = K_0 = C^1[a,b].$$

在本例中，两端的边界条件都是自然边界条件.

其他的边值问题，例如一端给出第一类边界条件，另一端给出第三类边界条件，可以类似得出它对应的变分问题.

以上讨论的几个变分问题中，函数集合 K 或 K_0 规定了函数 $y\in C^1[a,b]$，即 y 是在 $[a,b]$ 上具有一阶连续导数的函数. 事实上，这个条件可以改为较弱的条件，即要求 y 及其导数平方可积就可以了，就是要求 $\int_a^b\left[y^2+\left(\frac{\mathrm{d}y}{\mathrm{d}x}\right)^2\right]\mathrm{d}x$ 有意义. 这样所提的变分问题就存在惟一解. 这些有关微分方程的理论问题，在此不准备涉及.

下面把以上里茨变分问题和伽辽金变分问题的几个例子作一归纳. 设 K 是一个无限维的线性空间，它由满足某些条件的函数所组成. $F(\cdot)$ 是 K 上的一个泛函，即 K 到 $\mathbb{R}$ 的一个映射. 它满足：对任意的 $v,w\in K$，以及 $c\in\mathbb{R}$，有

$$F(v+w)=F(v)+F(w),$$
$$F(cw)=cF(w).$$

称 $F(\cdot)$ 是 K 上的一个**线性泛函**. 而 $D(\cdot,\cdot)$ 则是 $K\times K$ 上的一个**双线性泛函**，即它是 $K\times K$ 到 $\mathbb{R}$ 的一个映射，当 v,w 有一者固定时，$D(v,w)$ 是另一者的线性泛函. 双线性泛函 $D(\cdot,\cdot)$ 是对称的，即对一切 $v,w\in K$，有

$$D(v,w)=D(w,v).$$

同时又满足

$$D(v,v)\geqslant 0,\quad \forall v\in K.$$

至于里茨变分问题中的泛函 I，则满足

$$I(w)=\frac{1}{2}D(w,w)-F(w).$$

有了以上规定，**里茨变分问题**可以一般地描述为：

求 $y\in K$，使得对一切 $w\in K$ 有

$$I(y)\leqslant I(w). \tag{11.4.11}$$

伽辽金变分问题可以描述为：

求 $y \in K$，使得

$$D(y,w) - F(w) = 0 \qquad (11.4.12)$$

对一切 $w \in K$ 成立.

11.4.2 变分问题的近似计算

用变分方法解方程(11.4.1)，(11.4.3)，就是根据定理 11.4.1 描述的等价关系，求解里茨变分问题或伽辽金变分问题.

近似计算求解里茨变分问题(11.4.11)，先选取函数空间 K 的一个有限维子空间 S_N. S_N 的一组基是 $\{\varphi_1, \varphi_2, \cdots, \varphi_N\}$，对一切 $w_N \in S_N$，存在 $c_1, c_2, \cdots, c_N \in \mathbb{R}$，使

$$w_N = c_1\varphi_1 + c_2\varphi_2 + \cdots + c_N\varphi_N.$$

用 S_N 代替 K，可以建立一个里茨变分问题(11.4.11)的**近似变分问题**：

求 $y_N \in S_N$，使得对一切 $w_N \in S_N$ 有

$$I(y_N) \leqslant I(w_N). \qquad (11.4.13)$$

近似变分问题(11.4.13)的求解，只要把 $I(w_N)$ 列出：

$$\begin{aligned} I(w_N) &= \frac{1}{2}D\Big(\sum_{i=1}^{N} c_i\varphi_i, \sum_{j=1}^{N} c_j\varphi_j\Big) - F\Big(\sum_{i=1}^{N} c_i\varphi_i\Big) \\ &= \frac{1}{2}\sum_{i,j=1}^{N} D(\varphi_i, \varphi_j)c_i, \quad c_j - \sum_{i=1}^{N} c_i F(\varphi_i). \end{aligned}$$

然后对 $I(w_N)$ 求最小值，这只要令 $\dfrac{\partial I(w_N)}{\partial c_j} = 0 \ (j = 1, 2, \cdots, N)$，就得到 $c_1, c_2, \cdots, c_N$ 的一个线性方程组.

近似求解里茨变分问题的方法称为**里茨方法**. 它的求解过程是：

1. 选取 K 的子空间 S_N，它的一组基函数是 $\{\varphi_1, \varphi_2, \cdots, \varphi_N\}$.

2. 求解方程组

$$\sum_{j=1}^{N} D(\varphi_i,\varphi_j)c_j - F(\varphi_i) = 0 \quad (i=1,2,\cdots,N).$$

(11.4.14)

得到解$\tilde{c}_1,\tilde{c}_2,\cdots,\tilde{c}_N$.

3. 令 $y_N=\tilde{c}_1\varphi_1+\tilde{c}_2\varphi_2+\cdots+\tilde{c}_N\varphi_N$，这就是近似变分问题(11.4.13)的解.

伽辽金变分问题的**近似变分问题**是：

求 $y_N\in S_N$，使得

$$D(y_N,w_N)-F(w_N)=0 \qquad (11.4.15)$$

对一切 $w_N\in S_N$ 成立.

将近似变分问题的解 $y_N=\sum_{j=1}^{N}\tilde{c}_j\varphi_j$ 和 $w_N=\sum_{i=1}^{N}c_i\varphi_i$ 代入式(11.4.15)，得到

$$D(y_N,w_N)-F(w_N)=\sum_{i=1}^{N}\Big[\sum_{j=1}^{N}D(\varphi_i,\varphi_j)\,\tilde{c}_j-F(\varphi_i)\Big]c_i=0.$$

上式对任意的$(c_1,c_2,\cdots,c_N)$成立，即得到线性方程组

$$\sum_{j=1}^{N}D(\varphi_i,\varphi_j)\,\tilde{c}_j-F(\varphi_i)=0 \quad (i=1,2,\cdots,N).$$

它与方程组(11.4.14)是相同的. 这就是**伽辽金方法**的求解过程. 对于自伴随的边值问题，和里茨方法求解的结果是一样的.

例 11.4.4 用变分方法近似求解边值问题

$$\begin{cases}-\dfrac{d^2y}{dx^2}=x^2, & 0\leqslant x\leqslant 1,\\ y(0)=0, & y(1)=0.\end{cases}$$

解 本题变分问题中

$$K=\{y\mid y\in C^1[0,1],y(0)=0,y(1)=0\},$$

$$D(y,w)=\int_0^1\frac{dy}{dx}\frac{dw}{dx}dx,$$

$$F(w) = \int_0^1 x^2 w \mathrm{d}x.$$

取 K 中的子空间 S_N 为满足边界条件的多项式的集合，S_N 中的函数取为

$$w_N = x(1-x)(c_1 + c_2 x + \cdots + c_N x^{N-1}).$$

对于 $N=1$，$\varphi_1(x)=x(1-x)$，方程组(11.4.14)只有一个方程

$$D(\varphi_1, \varphi_1)c_1 - F(\varphi_1) = 0.$$

而 $D(\varphi_1, \varphi_1) = \int_0^1 \left(\frac{\mathrm{d}\varphi_1}{\mathrm{d}x}\right)^2 \mathrm{d}x = \frac{1}{3}$，$F(\varphi_1) = \int_0^1 x^3(1-x)\mathrm{d}x = \frac{1}{20}$.

解出 $\tilde{c}_1 = \frac{3}{20}$，所以边值问题近似解为

$$y_1 = \frac{3}{20}x(1-x).$$

对于 $N=2$，$\varphi_1(x)=x(1-x)$，$\varphi_2(x)=x^2(1-x)$，而

$$D(\varphi_1, \varphi_1) = \frac{1}{3}, \quad D(\varphi_1, \varphi_2) = \frac{1}{6}, \quad D(\varphi_2, \varphi_2) = \frac{2}{15},$$

$$F(\varphi_1) = \frac{1}{20}, \qquad F(\varphi_2) = \frac{1}{30}.$$

因此方程组(11.4.14)为

$$\begin{pmatrix} \frac{1}{3} & \frac{1}{6} \\ \frac{1}{6} & \frac{2}{15} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} \frac{1}{20} \\ \frac{1}{30} \end{pmatrix},$$

解出 $\tilde{c}_1 = \frac{1}{15}$，$\tilde{c}_2 = \frac{1}{6}$. 所以边值问题近似解为

$$y_2 = \frac{1}{30}x(1-x)(2+5x).$$

本例的精确解为 $y=\frac{x}{12}(1-x^3)$，下页表中列出几个点上 y，y_1 和 y_2 的值.

x	0	0.25	0.5	0.75	1
y_1	0	0.0281	0.0375	0.0281	0
y_2	0	0.0203	0.0375	0.0359	0
y	0	0.0205	0.0365	0.0361	0

11.5 有限元方法

解边值问题的**有限元方法**(finite element method)是一种变分方法. 在 11.4 节例 11.4.4 的变分近似计算中, S_N 取的是多项式函数的集合, 而在有限元方法中, S_N 的函数则是在$[a,b]$区间某种剖分下与节点有关的分段函数(一般是分段多项式)的集合, 这样更方便于实际计算.

在本节关于有限元方法一般叙述中, 我们主要以下列边值问题作为例子.

$$-\frac{\mathrm{d}}{\mathrm{d}x}\left(p(x)\frac{\mathrm{d}y}{\mathrm{d}x}\right)+q(x)y=f,\quad a\leqslant x\leqslant b, \tag{11.5.1}$$

$$y(a)=0,\quad p(b)\frac{\mathrm{d}y(b)}{\mathrm{d}x}+\alpha y(b)=g, \tag{11.5.2}$$

其中 $p\in C^1[a,b]$, $q,f\in C[a,b]$, $p(x)\geqslant p_0>0$, $q(x)\geqslant 0$, $\alpha\geqslant 0$.

边值问题(11.5.1), (11.5.2)对应的伽辽金变分问题是:

求 $y\in K$, 使得

$$D(y,w)-F(w)=0 \tag{11.5.3}$$

对一切 $w\in K$ 成立, 其中

$$K=\{y\mid y\in C^1[a,b], y(a)=0\}, \tag{11.5.4}$$

$$D(y,w)=\int_a^b\left[p(x)\frac{\mathrm{d}y}{\mathrm{d}x}\frac{\mathrm{d}w}{\mathrm{d}x}+q(x)yw\right]\mathrm{d}x+\alpha y(b)w(b), \tag{11.5.5}$$

$$F(w)=\int_a^b f(x)w\mathrm{d}x+gw(b). \tag{11.5.6}$$

11.5.1 线性元

在变分问题使用里茨方法或伽辽金方法求近似解的过程中，如果选用 S_N 为分段线性函数的集合，就构成最简单的有限元方法，称为使用线性元的有限元方法.

在$[a,b]$中插入节点$\{x_i\}_{i=0}^{N}$，满足

$$a = x_0 < x_1 < \cdots < x_N = b.$$

每个子区间 $e_i=[x_{i-1},x_i](i=1,2,\cdots,N)$称为**单元**，它的长度记为 $h_i=x_i-x_{i-1}$. 每个节点 x_i 对应分段线性插值基函数 $\varphi_i(x)$：

$$\varphi_0(x) = \begin{cases} \dfrac{x_1 - x}{h_1}, & 0 \leqslant x \leqslant x_1, \\ 0, & x_1 \leqslant x. \end{cases}$$

$$\varphi_i(x) = \begin{cases} 0, & x \leqslant x_{i-1}, \\ \dfrac{x - x_{i-1}}{h_i}, & x_{i-1} \leqslant x \leqslant x_i, \\ \dfrac{x_{i+1} - x}{h_{i+1}}, & x_i \leqslant x \leqslant x_{i+1}, \\ 0, & x_{i+1} \leqslant x, \end{cases} \tag{11.5.7}$$

$$i = 1,2,\cdots,N-1.$$

$$\varphi_N(x) = \begin{cases} 0, & x \leqslant x_{N-1}, \\ \dfrac{x - x_{N-1}}{h_N}, & x_{N-1} \leqslant x \leqslant x_N. \end{cases}$$

它们满足

$$\varphi_i(x_j) = \begin{cases} 1, & i = j, \\ 0, & i \neq j, \end{cases} \quad i,j = 0,1,\cdots,N.$$

设

$$S_{N+1} = \text{span}\{\varphi_0, \varphi_1, \cdots, \varphi_n\}, \tag{11.5.8}$$

即若 $w \in S_{N+1}$，则存在常数 $w_0, w_1, \cdots, w_N$ 使

$$w(x) = w_0\varphi_0(x) + w_1\varphi_1(x) + \cdots + w_N\varphi_N(x).$$

不难验证,上式的系数

$$w_i = w(x_i) \quad (i = 0,1,\cdots,N). \tag{11.5.9}$$

记

$$S_N = \{w \mid w \in S_{N+1}, w(x_0) = 0\}, \tag{11.5.10}$$

若 $w \in S_N$,则有

$$w(x) = w_1\varphi(x) + \cdots + w_N\varphi_N(x).$$

S_N 是变分问题(11.5.3)中函数空间 K 的一个 N 维子空间,由此列出近似变分问题:

求 $y \in S_N$,使得

$$\sum_{i=1}^{N}\int_{e_i}\left(p\frac{\mathrm{d}y}{\mathrm{d}x}\frac{\mathrm{d}w}{\mathrm{d}x} + qyw\right)\mathrm{d}x + \alpha y(x_N)w(x_N)$$

$$= \sum_{i=1}^{N}\int_{e_i} fw\,\mathrm{d}x + gw(x_N) \tag{11.5.11}$$

对一切 $w \in S_N$ 成立.

有了近似变分问题便可像上节那样得到线性方程组(11.4.14),但是一般有限元方法的具体计算过程如下描述.我们先设函数 $y \in S_{N+1}, w \in S_{N+1}$,最后才加进本质边界条件的约束,为此,记向量

$$\boldsymbol{w}_{e_i} = \begin{pmatrix} w_{i-1} \\ w_i \end{pmatrix}, \quad \boldsymbol{y}_{e_i} = \begin{pmatrix} y_{i-1} \\ y_i \end{pmatrix}.$$

$$\boldsymbol{N} = \left(\frac{x_i - x}{h_i}, \quad \frac{x - x_{i-1}}{h_i}\right) = (N_i(x), \quad M_i(x)),$$

$$\boldsymbol{B} = \left(-\frac{1}{h_i}, \quad \frac{1}{h_i}\right).$$

则在式(11.5.11)中

$$\int_{e_i}\left(p\frac{\mathrm{d}y}{\mathrm{d}x}\frac{\mathrm{d}w}{\mathrm{d}x} + gyw\right)\mathrm{d}x = \boldsymbol{w}_{e_i}^{\mathrm{T}}\boldsymbol{K}_{e_i}\boldsymbol{y}_{e_i} \quad (i = 1,2,\cdots,N),$$

其中,对 $i=1,2,\cdots,N-1$,

$$\begin{aligned}\boldsymbol{K}_{e_i} &= \int_{x_{i-1}}^{x_i} (p\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B} + q\boldsymbol{N}^{\mathrm{T}}\boldsymbol{N})\mathrm{d}x \\ &= \begin{pmatrix} k_{i-1,i-1}^{e_i} & k_{i-1,i}^{e_i} \\ k_{i,i-1}^{e_i} & k_{i,i}^{e_i} \end{pmatrix},\end{aligned} \tag{11.5.12}$$

$$\begin{cases} k_{i-1,i-1}^{e_i} = \dfrac{1}{h_i^2}\displaystyle\int_{x_{i-1}}^{x_i} p\mathrm{d}x + \int_{x_{i-1}}^{x_i} qN_i^2\mathrm{d}x, \\ k_{i-1,i}^{e_i} = k_{i,i-1}^{e_i} = -\dfrac{1}{h_i^2}\displaystyle\int_{x_{i-1}}^{x_i} p\mathrm{d}x + \int_{x_{i-1}}^{x_i} qN_iM_i\mathrm{d}x, \\ k_{i,i}^{e_i} = \dfrac{1}{h_i^2}\displaystyle\int_{x_{i-1}}^{x_i} p\mathrm{d}x + \int_{x_{i-1}}^{x_i} qM_i^2\mathrm{d}x, \end{cases} \tag{11.5.13}$$

这里 $i=1,2,\cdots,N-1$. 为了将近似变化问题(11.5.11)左端最后一项写进矩阵运算式子中,$\boldsymbol{K}_{e_N}$ 的右下角元素 $k_{NN}^{e_N}$ 应为

$$k_{NN}^{e_N} = \frac{1}{h_N^2}\int_{x_N}^{x_{N-1}} p\mathrm{d}x + \int_{x_{N-1}}^{x_N} qM_N^2\mathrm{d}x + \alpha, \tag{11.5.14}$$

而 $\boldsymbol{K}_{e_N}$ 的其他三个元素同式(11.5.13)前两式所示.

类似地处理式(11.5.11)右端的项,有

$$\int_{e_i} fw\mathrm{d}x = \boldsymbol{w}_{e_i}^{\mathrm{T}}\boldsymbol{F}_{e_i} \quad (i = 1,2,\cdots,N),$$

其中

$$\boldsymbol{F}_{e_i} = \begin{pmatrix} F_{i-1}^{e_i} \\ F_i^{e_i} \end{pmatrix}, \tag{11.5.15}$$

$$\begin{cases} F_{i-1}^{e_i} = \displaystyle\int_{x_{i-1}}^{x_i} N_i f\mathrm{d}x, \\ F_i^{e_i} = \displaystyle\int_{x_{i-1}}^{x_i} M_i f\mathrm{d}x, \end{cases} \quad i = 1,2,\cdots,N-1, \tag{11.5.16}$$

$$F_N^{e_N} = \int_{x_{N-1}}^{x_N} M_N f\mathrm{d}x + g, \tag{11.5.17}$$

而 $F_{N-1}^{e_N}$ 同式(11.5.16).

定义 11.5.1 由式(11.5.12)～式(11.5.14)确定的矩阵 $\boldsymbol{K}_{e_i}$ 称为用线性元求解边值问题(11.5.1),(11.5.2)的**单元刚度矩阵**(element stiffness matrix).

定义 11.5.2 由式(11.5.15)～式(11.5.17)确定的向量 $\boldsymbol{F}_{e_i}$ 称为用线性求解边值问题(11.5.1),(11.5.2)的**单元荷载向量**(element load vector).

记 $N+1$ 维的向量及 $(N+1)\times(N+1)$ 的矩阵:

$$\boldsymbol{y}=(y_0,y_1,\cdots,y_N)^{\mathrm{T}},$$

$$\boldsymbol{w}=(w_0,w_1,\cdots,w_N)^{\mathrm{T}},$$

$$\overline{\boldsymbol{K}}_{e_i}=\begin{pmatrix} & \vdots & \vdots & \\ \cdots & k_{i-1,i-1}^{e_i} & k_{i-1,i}^{e_i} & \cdots \\ \cdots & k_{i,i-1}^{e_i} & k_{i,i}^{e_i} & \cdots \\ & \vdots & \vdots & \end{pmatrix}\begin{matrix} \\ \text{第 } i-1 \text{ 行} \\ \text{第 } i \text{ 行} \\ \\ \end{matrix},$$

$$\qquad\qquad\text{第 } i-1 \text{ 列} \quad \text{第 } i \text{ 列}$$

$$\overline{\boldsymbol{F}}_{e_i}=(\cdots,F_{i-1}^{e_i},F_i^{e_i},\cdots)^{\mathrm{T}}.$$

在 $\overline{\boldsymbol{K}}_{e_i}$ 和 $\overline{\boldsymbol{F}}_{e_i}$ 中,省略处代表零元素.

定义 11.5.3 矩阵

$$\boldsymbol{K}=\sum_{i=1}^{N}\overline{\boldsymbol{K}}_{e_i} \tag{11.5.18}$$

称为用线性元求解边值问题(11.5.1)、(11.5.2)的**刚度矩阵**(stiffness matrix),或称**总刚度矩阵**.

定义 11.5.4 向量

$$\boldsymbol{F}=\sum_{i=1}^{N}\overline{\boldsymbol{F}}_{e_i} \tag{11.5.19}$$

称为用线性元求解边值问题(11.5.1)、(11.5.2)的**荷载向量**(load vector),或称**总荷载向量**.

可以求解一个与线性方程组(11.5.22)等价的 $N+1$ 阶线性方程组

$$\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & & \widetilde{\boldsymbol{K}} & \\ 0 & & & \end{pmatrix}\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_N \end{pmatrix} - \begin{pmatrix} 0 \\ \widetilde{\boldsymbol{F}} \\ \\ \end{pmatrix} = 0, \qquad (11.5.23)$$

它的系数矩阵是将 $\boldsymbol{K}$ 的第 1 行和第 1 列元素作了改变，同时 $\boldsymbol{F}$ 的第 1 个元素也变为 0，这样求解方程组(11.5.23)时，系数矩阵和 $\boldsymbol{K}$ 一样是 $N+1$ 阶的，比解方程组(11.5.23)要方便些.

例 11.5.6 用线性元方法解边值问题

$$\begin{cases} -\dfrac{\mathrm{d}^2 y}{\mathrm{d}x^2} = 2, \quad 0 \leqslant x \leqslant 1, \\ y(0) = 0, \quad y'(1) = 0. \end{cases}$$

解 本例中

$$D(y,w) = \int_0^1 \frac{\mathrm{d}y}{\mathrm{d}x}\frac{\mathrm{d}w}{\mathrm{d}x}\mathrm{d}x,$$

$$F(w) = 2\int_0^1 w\mathrm{d}x.$$

用节点 $x=0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1$ 将 $[0,1]$ 分为 4 个单元. $h_i = h = \frac{1}{4}$. 可得

$$\boldsymbol{K}_{e_i} = \frac{1}{h}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \quad (i = 1,2,3,4),$$

$$\boldsymbol{F}_{e_i} = h\begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad (i = 1,2,3,4).$$

$$\boldsymbol{K} = \sum_{i=1}^{4} \overline{\boldsymbol{K}}_{e_i} = \frac{1}{h}\begin{pmatrix} 1 & -1 & & & \\ -1 & 1+1 & -1 & & \\ & -1 & 1+1 & -1 & \\ & & -1 & 1+1 & -1 \\ & & & -1 & 1 \end{pmatrix},$$

$\boldsymbol{K}$ 是一个$(N+1)\times(N+1)$矩阵，它由 $\bar{\boldsymbol{K}}_{e_i}$ 迭加而成，而 $\bar{\boldsymbol{K}}_{e_i}$ 是 2×2 的单元刚度矩阵 $\boldsymbol{K}_{e_i}$ 的一种"放大"，即将 $\boldsymbol{K}_{e_i}$ 的四个元素放到$(N+1)\times(N+1)$矩阵 $\bar{\boldsymbol{K}}_{e_i}$ 相应的位置，而令 $\bar{\boldsymbol{K}}_{e_i}$ 其他的元素为 0，同理，对 $\boldsymbol{F}$ 的构成可作类似的分析.

考虑边界条件(11.5.2)中 $y(a)=0$，所以在向量 $\boldsymbol{y}$ 和 $\boldsymbol{w}$ 中，应取 $y_0=w_0=0$，用它们的分量作为系数作 $\varphi_0(x),\varphi_1(x),\cdots,\varphi_N(x)$ 的线性组合，得到的函数 $y(x)$ 和 $w(x)$ 都属于 S_N，这样，近似变分问题(11.5.11)可写成矩阵形式

$$(0,w_1,\cdots,w_N)\left(\boldsymbol{K}\begin{pmatrix}0\\y_1\\\vdots\\y_N\end{pmatrix}-\boldsymbol{F}\right)=0 \qquad (11.5.20)$$

对一切$(w_1,\cdots,w_N)$成立.或改写为

$$(w_1,\cdots,w_N)\left(\tilde{\boldsymbol{K}}\begin{pmatrix}y_1\\\vdots\\y_N\end{pmatrix}-\tilde{\boldsymbol{F}}\right)=0 \qquad (11.5.21)$$

对一切$(w_1,\cdots,w_N)$成立，其中 $\tilde{\boldsymbol{K}}$ 为 $\boldsymbol{K}$ 划去第 1 行和第 1 列所得到的 $N\times N$ 矩阵，$\tilde{\boldsymbol{F}}$ 为 $\boldsymbol{F}$ 划去第 1 个元素所得到的向量. 由$(w_1,\cdots,w_N)$的任意性，可得到线性方程组

$$\tilde{\boldsymbol{K}}\begin{pmatrix}y_1\\\vdots\\y_N\end{pmatrix}-\tilde{\boldsymbol{F}}=0. \qquad (11.5.22)$$

定理 11.5.5 对于边值问题(11.5.1)，(11.5.2)，线性方程组(11.5.22)的系数矩阵 $\tilde{\boldsymbol{K}}$ 是一个对称正定的矩阵，所以方程组(11.5.22)存在惟一的解向量$(y_1,\cdots,y_N)^{\mathrm{T}}$. 而 $\sum\limits_{j=1}^{N}y_i\varphi_i(x)$ 就是近似变分问题(11.5.11)的解，$y_i(i=1,2,\cdots,N)$是解在节点 x_i 的函数值.

$$F = \sum_{i=1}^{4} \bar{F}_{e_i} = h \begin{pmatrix} 1 \\ 1+1 \\ 1+1 \\ 1+1 \\ 1 \end{pmatrix}.$$

经过边界约束处理，得到线性方程组

$$4 \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \frac{1}{4} \begin{pmatrix} 2 \\ 2 \\ 2 \\ 1 \end{pmatrix}.$$

解方程组得

$$y_1 = \frac{7}{16}, \quad y_2 = \frac{12}{16}, \quad y_3 = \frac{15}{16}, \quad y_4 = 1.$$

11.5.2 高次元

线性元方法是将边值问题近似解用分段线性插值函数表示，而二次元方法则是用分段二次函数表示，其他高次元类似. 下面仿照线性元的方法，将插值基函数在每个单元 e_i 上做出，其他的运算则类似线性元方法.

关于二次元的插值基函数，为了计算方便，将区间$[x_{i-1}, x_i]$变换到变量 ξ 的区间$[-1,1]$. 在 $\xi \in [-1,1]$上作二次插值. 设插值节点为 $\xi=-1, \xi=0$ 和 $\xi=1$，则二次插值基函数为

$$\begin{cases} N_1(\xi) = \frac{1}{2}\xi(\xi-1), \\ N_2(\xi) = 1-\xi^2, \\ N_3(\xi) = \frac{1}{2}\xi(\xi+1). \end{cases} \tag{11.5.24}$$

记向量

$$\boldsymbol{N} = (N_1(\xi), \quad N_2(\xi) \quad N_3(\xi)),$$

$$\boldsymbol{y}_e = (y_1, \quad y_2, \quad y_3)^{\mathrm{T}},$$

则在 $\xi \in [-1,1]$上，二次插值函数表示为

$$y = \boldsymbol{N}\boldsymbol{y}_e.$$

例 11.5.7 用二次元方法解边值问题

$$\begin{cases} -\dfrac{\mathrm{d}^2 y}{\mathrm{d}x^2} = 1, & 0 \leqslant x \leqslant 1, \\ y(0) = 1, & \left(\dfrac{\mathrm{d}y}{\mathrm{d}x} + 2y\right)\Big|_{x=1} = 3. \end{cases}$$

解 本例的变分问题中

$$D(y,w) = \int_0^1 \frac{\mathrm{d}y}{\mathrm{d}x}\frac{\mathrm{d}w}{\mathrm{d}x}\mathrm{d}x + 2y(1)w(1),$$

$$F(w) = \int_0^1 w\mathrm{d}x + 3w(1).$$

把变量 x 的区间$[0,1]$平分为两个单元：

$$e_1 = \left[0, \frac{1}{2}\right], \quad e_2 = \left[\frac{1}{2}, 1\right],$$

其长度 $h_1 = h_2 = \dfrac{1}{2}$. 作两个变换

$$\xi = \frac{1}{h_i}[2x - (x_{i-1} + x_i)] \quad (i = 1,2),$$

它们分别将单元 e_1 和 e_2 变换变量 ξ 的标准单元$[-1,1]$，而且

$$\boldsymbol{B} = \frac{\mathrm{d}\boldsymbol{N}}{\mathrm{d}x} = \frac{1}{h_i}(2\xi - 1, \quad -4\xi, \quad 2\xi + 1).$$

可以计算单元刚度矩阵

$$\begin{aligned} \boldsymbol{K}_{e_i} &= \int_{e_i} \boldsymbol{B}^{\mathrm{T}}\boldsymbol{B}\mathrm{d}x \\ &= \int_{-1}^{1} \boldsymbol{B}^{\mathrm{T}}\boldsymbol{B}\frac{h_1}{2}\mathrm{d}\xi \end{aligned}$$

$$=\frac{1}{3}\begin{pmatrix} 14 & -16 & 2 \\ -16 & 32 & -16 \\ 2 & -16 & 14 \end{pmatrix}.$$

由于 $D(y,w)$ 中存在边界项 $2y(1)w(1)$，所以在计算 $\boldsymbol{K}_{e_2}$ 时，除和 $\boldsymbol{K}_{e_1}$ 相同外，对角线的第三个元素应该加上 2，得到

$$\boldsymbol{K}_{e_2}=\frac{1}{3}\begin{pmatrix} 14 & -16 & 2 \\ -16 & 32 & -16 \\ 2 & -16 & 20 \end{pmatrix}.$$

同理计算单元荷载向量为

$$\boldsymbol{F}_{e_1}=\int_{-1}^{1} f\boldsymbol{N}^{\mathrm{T}}\cdot\frac{h_1}{2}\mathrm{d}\xi=\frac{1}{3}\left(\frac{1}{4},\quad 1,\quad \frac{1}{4}\right)^{\mathrm{T}},$$

$$\boldsymbol{F}_{e_2}=\frac{1}{3}\left(\frac{1}{4},\quad 1,\quad \frac{37}{4}\right)^{\mathrm{T}}.$$

将单元的刚度矩阵及荷载向量分别对应迭加，得到

$$\boldsymbol{K}=\frac{1}{3}\begin{pmatrix} 14 & -16 & 2 & 0 & 0 \\ -16 & 32 & -16 & 0 & 0 \\ 2 & -16 & 28 & -16 & 2 \\ 0 & 0 & -16 & 32 & -16 \\ 0 & 0 & 2 & -16 & 20 \end{pmatrix},\quad \boldsymbol{F}=\frac{1}{3}\begin{pmatrix} \frac{1}{4} \\ 1 \\ \frac{1}{2} \\ 1 \\ \frac{37}{4} \end{pmatrix}.$$

考虑到边界条件 $y_0=1$，在约束处理后得到线性方程组

$$\frac{1}{3}\begin{pmatrix} 32 & -16 & 0 & 0 \\ -16 & 28 & -16 & 2 \\ 0 & -16 & 32 & -16 \\ 0 & 2 & -16 & 20 \end{pmatrix}\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix}=\frac{1}{3}\begin{pmatrix} 17 \\ -\frac{3}{2} \\ 1 \\ \frac{37}{4} \end{pmatrix}.$$

求解后得到 $y_0=1, y_1=\frac{39}{32}, y_2=\frac{11}{8}, y_3=\frac{47}{32}, y_4=\frac{3}{2}$.

关于三次的埃尔米特插值，将单元 e_i 变换到变量 ξ 的标准单元$[-1,1]$，并记

$$y_1 = y\Big|_{x=x_{i-1}}, \quad y_1' = \frac{\mathrm{d}y}{\mathrm{d}x}\Big|_{x=x_{i-1}},$$

$$y_2 = y\Big|_{x=x_i}, \quad y_2' = \frac{\mathrm{d}y}{\mathrm{d}x}\Big|_{x=x_i}.$$

令插值基函数

$$\begin{cases} N_1(\xi) = \frac{1}{4}(\xi^3 - 3\xi + 2), \\ N_2(\xi) = \frac{1}{4}(-\xi^3 + 3\xi + 2), \\ M_1(\xi) = \frac{h_i}{2} \cdot \frac{1}{4}(\xi^3 - \xi^2 - \xi + 1), \\ M_2(\xi) = \frac{h_i}{2} \cdot \frac{1}{4}(\xi^3 + \xi^2 - \xi + 1), \end{cases} \tag{11.5.25}$$

并记

$$\boldsymbol{N} = (N_1(\xi), \quad M_1(\xi), \quad N_2(\xi), \quad M_2(\xi)),$$

$$\boldsymbol{y}_{e_i} = (y_1, \quad y_1', \quad y_2, \quad y_2')^{\mathrm{T}},$$

则得到单元 e_i 上三次埃尔米特插值函数

$$y = \boldsymbol{N}\boldsymbol{y}_{e_i}. \tag{11.5.26}$$

用三次埃尔米特插值做有限元计算，每个节点上都有两个未知数 y_i, y_i'. 可以类似线性元的方法列出它们的线性方程组，解出 $y_0, y_0', y_1, y_1', \cdots, y_N, y_N'$. 近似解在每个单元如式(11.5.26)所示. 在整个区间$[a,b]$，近似解有一阶导数的连续性.

11.6 样条函数方法

讨论线性边值问题

$$y'' = p(x)y' + q(x)y + r(x), \quad a \leqslant x \leqslant b, \tag{11.6.1}$$

$$\alpha_0 y'(a) - \alpha_1 y(a) = \alpha, \quad \beta_0 y'(b) + \beta_1 y(b) = \beta. \tag{11.6.2}$$

其中 $\alpha_0\alpha_1 \geqslant 0, |\alpha_0| + |\alpha_1| \neq 0$；$\beta_0\beta_1 \geqslant 0, |\beta_0| + |\beta_1| \neq 0$.

引入节点 $x_0, x_1, \cdots, x_N$，对$[a,b]$作剖分

$$a = x_0 < x_1 < \cdots < x_n = b,$$

则$[a,b]$上以$\{x_0, x_1, \cdots, x_n\}$为节点的三次样条函数可以表示为

$$s(x) = e + f\cdot(x - x_0) + \frac{1}{2}g\cdot(x - x_0)^2 + \frac{1}{6}\sum_{j=0}^{n-1} d_j\cdot(x - x_j)_+^3, \tag{11.6.3}$$

其中 $e, f, g, d_0, d_1, \cdots, d_{n-1}$ 为 $n+3$ 个待定系数，函数

$$t_+ = \begin{cases} t, & \text{当 } t \geqslant 0, \\ 0, & \text{当 } t < 0. \end{cases}$$

用式(11.6.3)形式的函数 $s(x)$ 作为边值问题(11.6.1)，(11.6.2)的近似解. 不难对 $s(x)$ 求出其一、二阶导数，考虑在节点 x_i 上满足方程(11.6.1)，便有

$$\begin{aligned} g + \sum_{j=0}^{n-1} d_j(x_i - x_j)_+ = {} & p(x_i)\Big[f + g(x_i - x_0) + \frac{1}{2}\sum_{j=0}^{n-1} d_j(x_i - x_j)_+^2\Big] + \\ & q(x_i)\Big[e + f(x_i - x_0) + \frac{1}{2}g(x_i - x_0)^2 + \\ & \frac{1}{6}\sum_{j=0}^{n-1} d_j(x_i - x_j)_+^3\Big] + r(x_i), \\ & i = 0, 1, \cdots, n. \end{aligned} \tag{11.6.4}$$

同理，由边界条件(11.6.2)得到

$$\alpha_0 f - \alpha_1 e = \alpha, \tag{11.6.5}$$

$$\begin{aligned} & \beta_0\Big[f + g(x_n - x_0) + \frac{1}{2}\sum_{j=0}^{n-1} d_j(x_n - x_j)_+^2\Big] + \\ & \beta_1\Big[e + f(x_n - x_0) + \frac{1}{2}g(x_n - x_0)^2 + \frac{1}{6}\sum_{j=0}^{n-1} d_j(x_n - x_j)_+^3\Big] = \beta. \end{aligned} \tag{11.6.6}$$

式(11.6.4)～式(11.6.6)是 $n+3$ 个未知数 $e,f,g,d_0,\cdots,d_{n-1}$ 的 $n+3$ 个方程，解这个线性方程组，代入式(11.6.3)就得到边值问题的近似解 $s(x)$.

例 11.6.1 用样条函数方法解边值问题

$$\begin{cases} y''+y+1=0, & 0\leqslant x\leqslant 1, \\ y(0)=0, & y(1)=0. \end{cases}$$

解 把区间[0,1]三等分，节点为 $0,\frac{1}{3},\frac{2}{3}$ 和 1. 式(11.6.3)中函数 $s(x)$ 有 6 个待定系数 e,f,g,d_0,d_1,d_2. 由边界条件 $y(0)=0$，即得 $e=0$，所以设

$$s(x)=fx+gx^2+d_0x_+^3+d_1\left(x-\frac{1}{3}\right)_+^3+d_2\left(x-\frac{2}{3}\right)_+^3, \tag{11.6.7}$$

求出 $s'(x)$ 及 $s''(x)$，代入边值问题中的微分方程，并取 x 为 $0,\frac{1}{3},\frac{2}{3},1$，再加上边界条件 $y(1)=0$，可得到线性方程组

$$\begin{cases} 2g=-1, \\ 55d_0+57g+9f=-27, \\ 55d_1+116d_0+66g+18f=-27, \\ 55d_2+116d_1+189d_0+81g+275f=-27, \\ d_2+8d_1+27d_0+27g+27f=0. \end{cases}$$

解出

$$f=0.540814, \qquad g=-0.5,$$

$$d_0=-0.061224, \quad d_1=0.061224, \quad d_2=0.061224.$$

将其代入式(11.6.7)即得边值问题近似解.

12 偏微分方程数值解法

12.1 椭圆型方程的有限差分方法

12.1.1 典型问题

椭圆型方程(elliptic equations)中最简单的典型方程是**拉普拉斯**(Laplace)**方程**

$$Lu = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \tag{12.1.1}$$

和泊松(Poisson)**方程**

$$Lu = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x,y). \tag{12.1.2}$$

椭圆型方程的主要定解问题是边值问题(boundary value problems),即要求定出未知函数 u,使得它在某个区域 Ω 内满足微分方程(12.1.1)或(12.1.2),并在区域的边界$\partial\Omega$上满足给定的**边界条件**(boundary conditions). 边界条件可以由下面三种方式给出:

$$u\Big|_{\partial\Omega} = \alpha(x,y), \tag{12.1.3}$$

$$\frac{\partial u}{\partial n}\Big|_{\partial\Omega} = \beta(x,y), \tag{12.1.4}$$

$$\left(\frac{\partial u}{\partial n} + ku\right)\Big|_{\partial\Omega} = \gamma(x,y), \tag{12.1.5}$$

其中 n 表示外法线方向,$k=k(x,y)\geqslant 0$,边界条件(12.1.3)、(12.1.4)和(12.1.5)分别称为第一类、第二类和第三类边界条件.

12.1.2 网格和差分近似

设 Ω 为 Oxy 平面上的一有界区域，$\partial\Omega$ 为其边界(见图 12.1)．取定沿 x 轴方向步长为 h_1，y 轴方向步长为 h_2，作两族与坐标轴平行的直线：

$$x = ih_1 \quad (i = 0, \pm 1, \pm 2, \cdots),$$

$$y = jh_2 \quad (j = 0, \pm 1, \pm 2, \cdots).$$

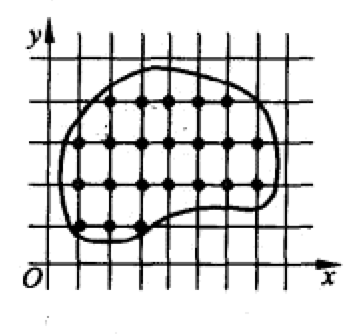


图 12.1

两族直线的交点(ih_1, jh_2)称为**网格点**或**节点**，记为(x_i, y_j)．仅考虑属于 $\Omega \cup \partial\Omega$ 的网格点．如果两个网格点(x_i, x'_j)和(x'_i, y'_j)满足下面关系式

$$\left|\frac{x_i - x'_i}{h_1}\right| + \left|\frac{y_j - y'_j}{h_2}\right| = 1,$$

那么称这两个网格点是**相邻的**．

如果一个节点的 4 个相邻的节点都属于 $\Omega \cup \partial\Omega$，则称此节点为**内部节点**，或简称为**内点**．全部内点的全体记作 Ω_h．如果一个节点的 4 个相邻的节点至少有一个不属于 $\Omega \cup \partial\Omega$，则称其为**边界节点**，或简称为界点．全体界点的集合记为$\partial\Omega_h$．在图 12.1 中，以"•"表示界点，以"◦"表示内点．利用泰勒级数展开

$$\frac{1}{2h_1}[u(x_{i+1}, y_j) - u(x_{i-1}, y_j)]$$

$$= \frac{\partial u(x_i, y_j)}{\partial x} + \frac{1}{6}h_1^2 \frac{\partial^3 u(x_i, y_j)}{\partial x^3} + O(h_1^4),$$

有

$$\left(\frac{\partial u}{\partial x}\right)_{ij} = \frac{\partial u(x_i, y_j)}{\partial x} \approx \frac{u(x_{i+1}, y_j) - u(x_{i-1}, y_j)}{2h_1}. \tag{12.1.6}$$

同样地，有

$$\left(\frac{\partial u}{\partial y}\right)_{ij}=\frac{\partial u(x_i,y_j)}{\partial y}\approx\frac{u(x_i,y_{j+1})-u(x_i,y_{j-1})}{2h_2}.$$

(12.1.7)

对于二阶偏导数,同样可用泰勒级数展开获得近似表达式,并且容易求得

$$\begin{cases}\left(\dfrac{\partial^2 u}{\partial x^2}\right)_{ij}=\dfrac{\partial^2 u(x_i,y_j)}{\partial x^2}\\ \qquad\approx\dfrac{u(x_{i+1},y_j)-2u(x_i,y_j)+u(x_{i-1},y_i)}{h_1^2},\\ \left(\dfrac{\partial^2 u}{\partial y^2}\right)_{ij}=\dfrac{\partial^2 u(x_i,y_j)}{\partial y^2}\\ \qquad\approx\dfrac{u(x_i,y_{j+1})-2u(x_i,y_j)+u(x_i,y_{j-1})}{h_2^2}.\end{cases}$$

(12.1.8)

12.1.3 差分格式的构造方法

由泊松方程(12.1.2)的有限差分格式(finite difference scheme)构造方法,直接可以得到拉普拉斯方程(12.1.1)的有限差分格式.因此,仅讨论泊松方程的有限差分格式.有限差分格式也称为**有限差分方法**.

取正方形网格:$h=h_1=h_2$,任取一内点(x_i,y_j),利用式(12.1.8)可以由泊松方程直接得到一近似表示式

$$\frac{u(x_{i+1},y_j)-2u(x_i,y_j)+u(x_{i-1},y_j)}{h^2}+$$

$$\frac{u(x_i,y_{j+1})-2u(x_i,y_j)+u(x_i,y_{j-1})}{h^2}\approx -f(x_i,y_j).$$

用u_{ij}表示$u(x_i,y_j)$的近似值,则由上式可以得到逼近泊松方程的**有限差分格式**,简称**差分格式**.

$$L_h u=\frac{u_{i+1,j}-2u_{ij}+u_{i-1,j}}{h^2}+\frac{u_{i,j+1}-2u_{ij}+u_{i,j-1}}{h^2}$$

$$=-f_{ij},\qquad(12.1.9)$$

其中 $f_{ij}=f(x_i,y_j)$.

上述构造差分格式(12.1.9)的办法是用泰勒级数展开,这是最方便的方法.差分格式(12.1.9)中只出现 u 在(x_i,y_i)及其 4 个邻点上的值,所以称其为**五点差分格式**(five point difference scheme).

有限体积法(finite volume method)是推导偏微分方程的有限差分格式的重要方法.下面用于推导泊松方程的五点差分格式(此仅为说明方法).

设 $P=(x_i,y_j)\in\Omega_h$,其 4 个邻点为 $Q_1=(x_{i+1},y_j)$,$Q_2=(x_i,y_{j+1})$,$Q_3=(x_{i-1},y_j)$和 $Q_4=(x_i,y_{j-1})$.令 N_i 为 PQ_i 的中点,$i=1,2,3,4$(见图 12.2).在阴影区域 D_{ij}上对泊松方程(12.1.2)两边进行积分,有

$$\iint\limits_{D_{ij}}\frac{\partial^2 u}{\partial x^2}\mathrm{d}x\mathrm{d}y=\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}}\left(\frac{\partial u}{\partial x}\bigg|_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}}\right)\mathrm{d}y,$$

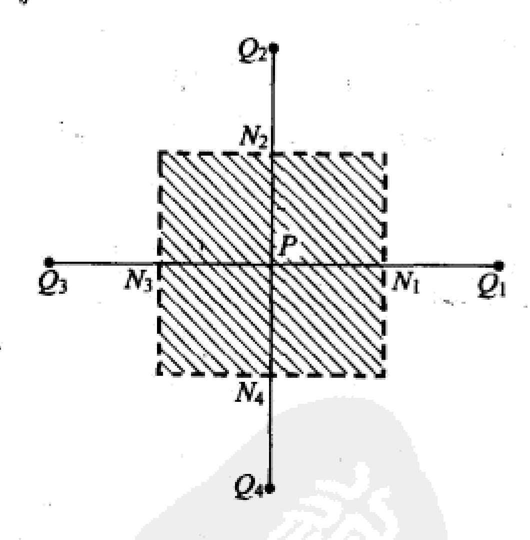


图 12.2

其中 $x_{i\pm\frac{1}{2}}=\frac{1}{2}(x_i+x_{i\pm1})$,$y_{j\pm\frac{1}{2}}=\frac{1}{2}(y_i,y_{j\pm1})$.

利用中点求积公式有

$$\iint\limits_{D_{ij}} \frac{\partial^2 u}{\partial x^2} \mathrm{d}x\mathrm{d}y \approx \left[\frac{\partial u(x_{i+\frac{1}{2}}, y_j)}{\partial x} - \frac{\partial u(x_{i-\frac{1}{2}}, y_j)}{\partial x}\right]h,$$

把上式右边的微商用中心差商来代替，就得到

$$\iint\limits_{D_{ij}} \frac{\partial^2 u}{\partial x^2} \mathrm{d}x\mathrm{d}y \approx u_{i+1,j} - 2u_{ij} + u_{i-1,j}.$$

同理，有

$$\iint\limits_{D_{ij}} \frac{\partial^2 u}{\partial y^2} \mathrm{d}x\mathrm{d}y \approx u_{i,j+1} - 2u_{ij} + u_{i,j-1}.$$

此外

$$\iint\limits_{D_{ij}} f(x,y)\mathrm{d}x\mathrm{d}y \approx f_{ij}h^2.$$

综合上述各近似式，可以得到逼近泊松方程的差分格式

$$L_h u = \frac{1}{h^2}[(u_{i+1,j} - 2u_{ij} + u_{i-1,j}) + (u_{i,j+1} - 2u_{ij} + u_{i,j-1})]$$
$$= -f_{ij}.$$

这就是前面推导出来的五点差分格式. 利用有限体积法来构造变系数方程、间断系数方程和非线性方程的有限差分格式较为方便.

12.1.4 常用的有限差分格式

五点差分格式(12.1.9)是经常使用的，它的节点排列以(x_i, y_j)为中心，其余4个点是(x_i, y_j)的上、下、左、右四个相邻的节点(见图12.3(a)). 如果采用图12.3(b)的节点，也可以得到逼近泊松方程的差分格式

$$L_h^{(1)} u = \frac{1}{2h^2}(u_{i+1,j+1} + u_{i+1,j-1} + u_{i-1,j+1} + u_{i-1,j-1} - 4u_{ij}) = -f_{ij}. \tag{12.1.10}$$

利用差分格式(12.1.9)和式(12.1.10)进行线性组合可以得

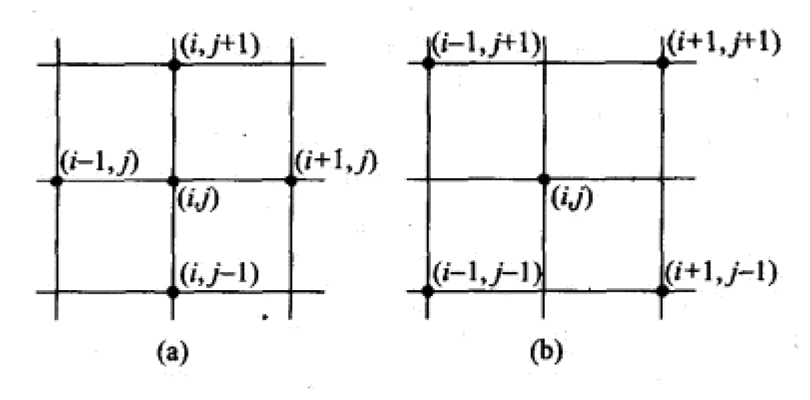


图 12.3

到逼近泊松方程的另一差分格式

$$L_h^{(2)}u = \frac{1}{6h^2}[4(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{ij}) + (u_{i+1,j+1} + u_{i+1,j-1} + u_{i-1,j+1} + u_{i-1,j-1} - 4u_{ij})]$$
$$= -\left[f_{ij} + \frac{1}{12}(f_{i,j+1} + f_{i,j-1} + f_{i+1,j} + f_{i-1,j} - 4f_{ij})\right]. \tag{12.1.11}$$

由于差分格式(12.1.9)和式(12.1.10)都仅采用五个节点，而差分格式(12.1.11)采用了九个节点，因此一般也称差分格式(12.1.11)为**九点差分格式**. 对于一个差分格式，其基本的要求是差分方程逼近微分方程，这种逼近的近似程度的好坏可以用**截断误差**(truncation error)来描述.

设 $u=u(x,y)$ 是泊松方程(12.1.2)的充分光滑的解，L_h 是由式(12.1.9)定义的**差分算子**(difference operator)，称

$$R_{ij}(u) = L_h u(x_i,y_j) - Lu(x_i,y_j) \tag{12.1.12}$$

为差分格式(12.1.9)的截断误差.

实际计算截断误差时，只要将 $L_h u(x_i,y_j)$ 的各项在 (x_i,y_j) 展成泰勒级数即可通过初等计算可以得到，五点差分格式(12.1.9)和(12.1.10)的截断误差是 $O(h^2)$，而九点差分格式(12.1.11)的截

断误差为 $O(h^4)$. 由于九点差分格式(12.1.11)截断误差的阶高,所以经常称其为**高阶格式**. 显然,它更精确地逼近泊松方程. 在实际计算中,如果精度要求不是很高,以采用五点差分格式较为合适.

12.1.5 边界条件的处理

在对内点列出差分方程后,为了求解,还必须对边界条件进行处理以给出边界上的关系式.

对第一类边界条件

$$u|_{\partial\Omega} = \alpha(x,y), \quad (x,y)\in\partial\Omega,$$

在构造网格时,$\partial\Omega_h$ 通常都不与 $\partial\Omega$ 重合,可能有少数边界节点落在 $\partial\Omega$ 上,而大部分边界节点不落在 $\partial\Omega$ 上,由此而产生边界条件的转换.

(1) 直接转移法. 若边界节点 (x_i, y_j) 正好落在 $\partial\Omega$ 上,如图 12.4 所示,则 $u_{ij}=\alpha(x_i, y_j)$. 如果 (x_i, y_j) 不在 $\partial\Omega$ 上,此时 $\partial\Omega$ 与网格线交于 P 和 Q 两点,则取与点 (x_i, y_j) 距离最近的点 $\alpha(x, y)$ 的值为 u_{ij}.

(2) 线性插值. 如图 12.5 所示,对于网格点 P 可用 R, Q 两点作线性插值

$$u(P)=\frac{h}{h+\delta}u(R)+\frac{\delta}{h+\delta}u(Q),$$

其中 $\delta=RP<h$.

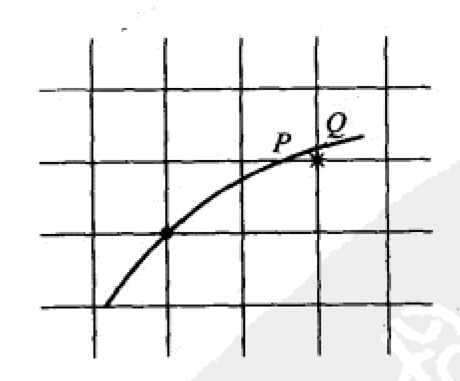


图 12.4

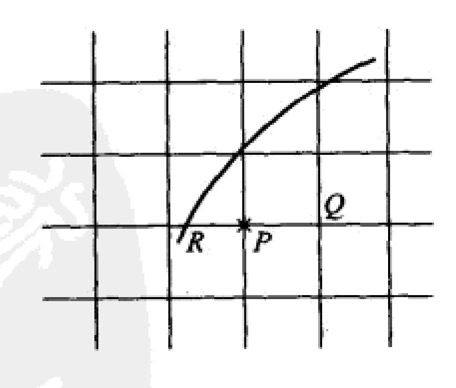


图 12.5

由于第二类边界条件可以看作第三类边界条件的特殊情况，所以仅处理第三类边界条件

$$\left(\frac{\partial u}{\partial n}+ku\right)\bigg|_{\partial\Omega}=\gamma(x,y).$$

仍假定正方形网格. 有以下三种情况：

(1) 点 P 在边界 $\partial\Omega$ 上，且外法向 $\boldsymbol{n}$ 与坐标轴平行(见图 12.6)

$$\frac{u(P)-u(Q)}{h}+ku(P)=\gamma(P).$$

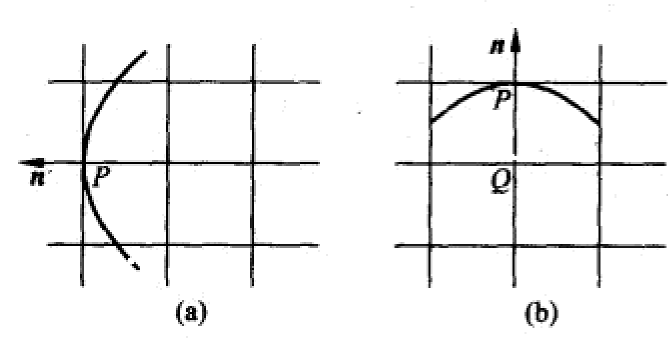


图 12.6

(2) 点 P 在边界 $\partial\Omega$ 上，但外法向 $\boldsymbol{n}$ 与坐标轴不平行(见图 12.7).

$$\frac{u(P)-u(Q)}{h}\cos\alpha+\frac{u(P)-u(R)}{h}\cos\beta+ku(P)=\gamma(P).$$

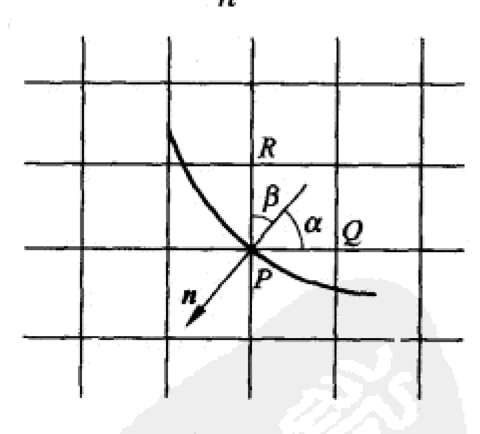


图 12.7

(3) 点 P 不在边界 $\partial\Omega$ 上(见图 12.8).

$$\frac{u(P)-u(Q)}{h}\cos\alpha+\frac{u(P)-u(R)}{h}\cos\beta+ku(P)=\gamma(P^*),$$

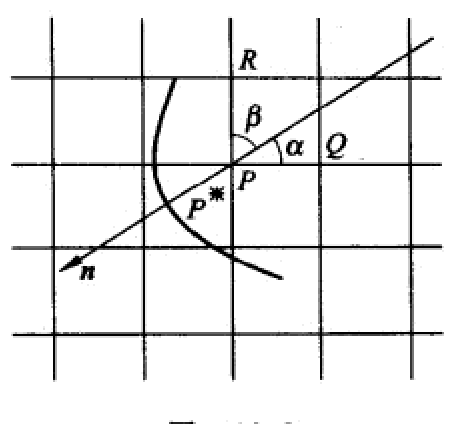


图 12.8

其中 P^* 为距 P 最近的$\partial\Omega$ 上一点.

例 12.1.1 在 $\Omega=\{(x,y)\,|\,0<x<0.5, 0<y<0.5\}$ 内考虑拉普拉斯方程

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=0,$$

边界条件为

$$u(0,y)=0,\qquad u(x,0)=0,$$
$$u(x,0.5)=200x,\quad u(0.5,y)=200y.$$

用正方形网格 $h=0.125$ 来求解上述问题.

解 采用差分格式(12.1.9)

$$4u_{ij}-u_{i+1,j}-u_{i-1,j}-u_{i,j+1}-u_{i,j-1}=0,$$
$$i=1,2,3,\quad j=1,2,3,$$

用网格点来表明这些量(网格点分布见图 12.9),那么方程可以写为

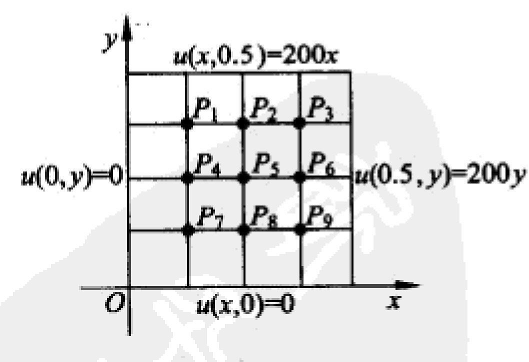


图 12.9

$$4u_1 - u_2 - u_4 = u_{03} + u_{14},$$
$$4u_2 - u_3 - u_1 - u_5 = u_{24},$$
$$4u_3 - u_2 - u_6 = u_{43} + u_{34},$$
$$4u_4 - u_5 - u_1 - u_7 = u_{02},$$
$$4u_5 - u_6 - u_4 - u_2 - u_8 = 0,$$
$$4u_6 - u_5 - u_3 - u_9 = u_{42},$$
$$4u_7 - u_8 - u_4 = u_{01} + u_{10},$$
$$4u_8 - u_9 - u_7 - u_5 = u_{20},$$
$$4u_9 - u_8 - u_6 = u_{30} + u_{41}.$$

方程的右边项可从边界条件的直接转移法得到：

$$u_{10} = u_{20} = u_{30} = u_{01} = u_{02} = u_{03} = 0,$$
$$u_{14} = u_{41} = 25, \quad u_{24} = u_{42} = 50, \quad u_{34} = u_{43} = 75,$$

从而可得线性方程组

$$\begin{pmatrix} 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 4 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 4 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 4 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \end{pmatrix} = \begin{pmatrix} 25 \\ 50 \\ 150 \\ 0 \\ 0 \\ 50 \\ 0 \\ 0 \\ 25 \end{pmatrix}.$$

用高斯-赛德尔方法求解上述线性代数方程组，其结果如下：

i	1	2	3	4	5	6	7	8	9
u_i	18.75	37.50	56.25	12.50	25.00	37.50	6.25	12.50	18.75

例 12.1.2 在区域 $\Omega=\{(x,y)\,|\,0\leqslant x\leqslant 2, 0\leqslant y\leqslant 1\}$ 内考虑拉普拉斯方程

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=0,$$

边界条件如图 12.10 所示（出现第二类边界条件）. 用正方形网格

$h=0.5$ 求解上述问题.

解 用差分格式(12.1.9)来求解,网格节点见图 12.10,对于内点 4,5,6 有

$$4:\quad 4u_4-u_1-u_5=\frac{1}{2},$$
$$5:\quad 4u_5-u_2-u_4-u_6=1,$$
$$6:\quad 4u_6-u_3-u_5=\frac{5}{2}.$$

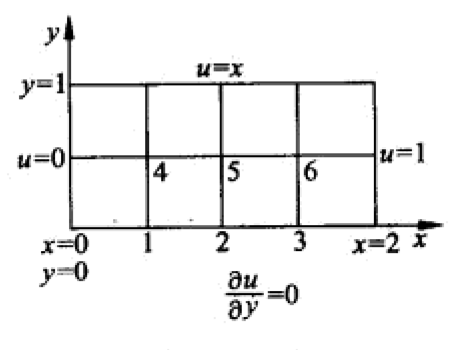


图 12.10

对于点 1,2,3 需要将相应点上的$\frac{\partial u}{\partial y}=0$进行离散,以点 2 为例,

$$\left.\frac{\partial u}{\partial y}\right|_2=\frac{u_5-u_P}{2h}=0,$$

其中 P 为虚构网格点,并有 $u_P=u_5$,这样,在点 2 利用式(12.1.9)可得

$$2:\quad 4u_2-2u_5-u_1-u_3=0.$$

同样处理点 1,3,有

$$1:\quad 4u_1-2u_4-u_2=0,$$
$$3:\quad 4u_3-2u_6-u_2=1.$$

6 个差分方程写成矩阵形式方程为

$$\begin{pmatrix} 4 & -1 & 0 & -2 & 0 & 0 \\ -1 & 4 & -1 & 0 & -2 & 0 \\ 0 & -1 & 4 & 0 & 0 & -2 \\ -1 & 0 & 0 & 4 & -1 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & -1 & 0 & -1 & 4 \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{pmatrix}=\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0.5 \\ 1 \\ 2.5 \end{pmatrix}.$$

用高斯-赛德尔迭代得

$$u_1 = 0.401786,$$
$$u_2 = 0.750000,$$
$$u_3 = 0.973214,$$
$$u_4 = 0.428571,$$
$$u_5 = 0.812500,$$
$$u_6 = 1.071429.$$

上述例子说明，利用了内点的差分方程和边界条件的离散后，一般都可得到线性代数方程组．对此可用雅可比迭代法、高斯-赛德尔迭代法、逐次超松弛方法等进行求解，并且还可以利用许多加速收敛的方法.

12.2 双曲型方程的差分方法

12.2.1 典型问题

最简单的**双曲型方程**(hyperbolic equations)是**对流方程**(convection equations)：

$$\frac{\partial u}{\partial t}+a\frac{\partial u}{\partial x}=0,\quad -\infty<x<\infty,\quad t>0. \tag{12.2.1}$$

设给定初始条件

$$u(x,0)=\varphi(x),\quad -\infty<x<\infty, \tag{12.2.2}$$

容易验证对流方程(12.2.1)的解为

$$u(x,t)=\varphi(x-at),\quad -\infty<x<\infty,\quad t\geqslant 0. \tag{12.2.3}$$

这意味着在 Oxt 平面上沿每条直线 $x-at=C$(C 为常数)，u 值保持不变，这种直线就是**特征线**(characteristic line)(见图 12.11).

当 $a>0$ 时，特征线向左倾斜；当 $a<0$ 时，特征线向右倾斜. 在任一点(x,t)上的 u 值仅依赖于初始函数 φ 在 $\xi=x-at$ 的值，ξ 就是通过(x,t)的特征线与 x 轴交点的横坐标. 故解 u 关于初值的**依赖**

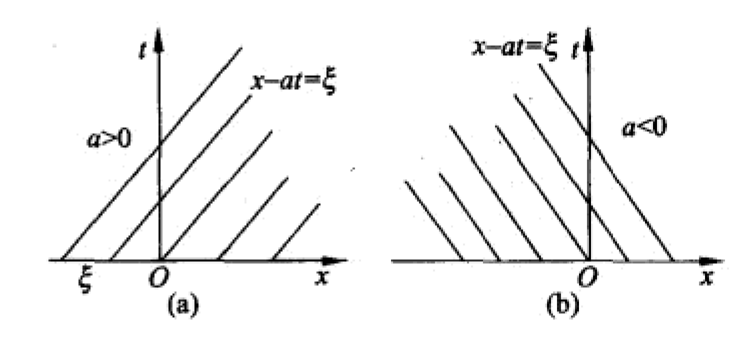


图 12.11

区域(domain of dependence)用单个点 ξ 来表示,如图 12.11 所示.

如果定解区域 $\Omega=\{(x,t)\mid 0\leqslant x\leqslant l, t\geqslant 0\}$,则要给出适当的边界条件.由于对流方程的解在其特征线上是不变的,所以边界条件不能任意给定,如果 $a>0$,只能在左边界给条件;如果 $a<0$,只能在右边界给条件.

双曲型方程的另一个典型例子是**波动方程**(wave equation):

$$\frac{\partial^2 u}{\partial t^2}=a^2\frac{\partial^2 u}{\partial x^2},\quad -\infty<x<\infty,\quad t>0,\tag{12.2.4}$$

其中 $a>0$.给定初始条件

$$\begin{cases}u(x,0)=\varphi(x), & -\infty<x<\infty,\\ \dfrac{\partial u(x,0)}{\partial t}=\psi(x), & -\infty<x<\infty,\end{cases}\tag{12.2.5}$$

则初值问题(12.2.4),(12.2.5)的解可由达朗贝尔(D'Alembert)公式给出:

$$u(x,t)=\frac{\varphi(x+at)+\varphi(x-at)}{2}+\frac{1}{2a}\int_{x-at}^{x+at}\psi(\xi)\mathrm{d}\xi.\tag{12.2.6}$$

由达朗贝尔公式不难看出,初值问题(12.2.4),(12.2.5)的解 u 在点 (x^*,t^*) 的值 $u(x^*,t^*)$,只依赖于 x 轴上由 x^*-at^* 到 x^*+at^* 之间的初值 φ 和 ψ,因此区间 $[x^*-at^*,x^*+at^*]$ 称为点 (x^*,t^*) 的**依赖区间**(或称依赖区域),见图 12.12.

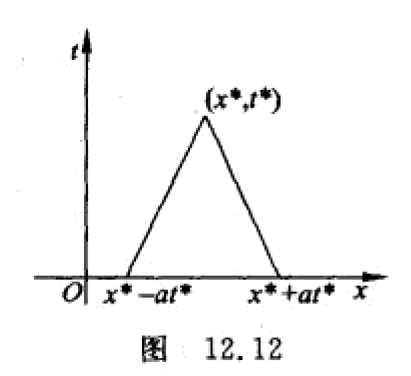


图 12.12

对于波动方程(12.2.4),还可以有初边值混合问题,设混合问题的求解区域为

$$\Omega=\{(x,t)\mid 0\leqslant x\leqslant l,0\leqslant t\},$$

由此波动方程及其初始条件的空间变量 x 都限于区间 $[0,l]$.

(1) 对第一类边界条件,有

$$u\big|_{x=0}=\Phi_0(t),u\big|_{x=l}=\Phi_1(t),\quad t\geqslant 0. \tag{12.2.7}$$

(2) 对第三类边界条件,有

$$\begin{cases}\left(\dfrac{\partial u}{\partial x}+\eta_0(t)u\right)\Big|_{x=0}=F_0(t), & t\geqslant 0,\\ \left(\dfrac{\partial u}{\partial x}+\eta_1(t)u\right)\Big|_{x=l}=F_1(t), & t\geqslant 0.\end{cases} \tag{12.2.8}$$

12.2.2 差分格式

求解对流方程和波动方程的初值问题时,求解区域为

$$\Omega=\{(x,t)\mid -\infty<x<\infty,t\geqslant 0\},$$

即为 Oxt 的上半平面.为建立差分方程,首先要剖分网格,设 h 为 x 轴方向的步长(空间步长),τ 为 t 轴方向的步长(时间步长).令

$$x_j=jh\quad(j=0,\pm1,\pm2,\cdots),\quad t_n=n\tau\quad(n=0,1,2,\cdots),$$

这是二族平行于 x 轴和 t 轴的直线,它们构成了 Oxt 上半平面的网格,如图 12.13.

一般情况下,h 可预先给定,τ 则根据不同的差分格式而定.

构造差分格式最简单的方法,是用差商代替微商.以对流方程(12.2.1)为例,构造两个常见格式.假定 $a>0$,注意到

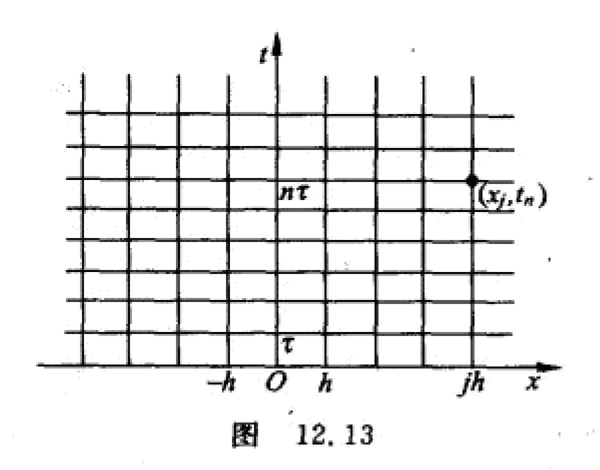


图 12.13

$$\frac{\partial u(x_j,t_n)}{\partial t}=\frac{u(x_j,t_{n+1})-u(x_j,t_n)}{\tau}+O(\tau),$$
$$\frac{\partial u(x_j,t_n)}{\partial x}=\frac{u(x_{j+1},t_n)-u(x_j,t_n)}{h}+O(h),$$
$$\frac{\partial u(x_j,t_n)}{\partial x}=\frac{u(x_j,t_n)-u(x_{j-1},t_n)}{h}+O(h),$$

立即就可得出逼近对流方程(12.2.1)的两个差分格式:

$$\frac{u_j^{n+1}-u_j^n}{\tau}+a\,\frac{u_{j+1}^n-u_j^n}{h}=0, \qquad (12.2.9)$$

$$\frac{u_j^{n+1}-u_j^n}{\tau}+a\,\frac{u_j^n-u_{j-1}^n}{h}=0, \qquad (12.2.10)$$

其中 u_j^n 是在网格点上取值的函数.

当由第 n 个时间层(在 $t=n\tau$ 的一排节点)推进到 $n+1$ 层时,公式(12.2.9)和公式(12.2.10)都提供了逐点计算 u_j^{n+1} 的明显表达式,因此称它们为**显式格式**(explicit scheme).在公式(12.2.9)中,在推算 $n+1$ 层时只用到第 n 层的数据,前后联系到两个时间层次,所以格式(12.2.9)称为**两层差分格式**(two-level difference scheme).同样,差分格式(12.2.10)也是两层差分格式.

给出的差分格式是否逼近到微分方程,以及逼近微分方程的程度如何都可以用截断误差来描述.所谓截断误差就是把微分方程的光滑解代入到差分方程中并做泰勒级数展开后所得到的余

项，容易验证，差分方程(12.2.9)和(12.2.10)的截断误差是 $O(\tau)+O(h)$，一般简写为 $O(\tau+h)$. 如果步长 $h,\tau\to 0$，截断误差也趋于 0，那么称差分方程与微分方程是相容的. **相容性**(consistency)表示差分方程"收敛"于微分方程，这是用差分方程求解的必备条件. 由于差分格式是多种多样的，因此截断误差也将不同. 如果截断误差 $E=O(\tau^p+h^q)$，则称差分格式对时间(τ)为 p 阶精度，对空间(h)为 q 阶精度. 如果 $p=q$，则称差分格式是 p 阶精度的. 显然差分格式(12.2.9)、(12.2.10)都是一阶精度差分格式.

对于一个差分格式，要弄清两个问题：

(1) 当步长 $h,\tau\to 0$ 时，差分格式的解是否收敛到微分方程的初值问题的解；

(2) 计算中产生的误差，在以后的计算中是无限增加还是可以控制.

第一个问题称为**收敛性**(convergence)**问题**，第二个问题称为**稳定性**(stability)**问题**.

考虑差分格式(12.2.9)的收敛性问题. 先把式(12.2.10)变形

$$u_j^n=(1-a\lambda)u_j^{n-1}+a\lambda u_{j-1}^{n-1}$$

其中 $\lambda=\tau/h$，一般称其为**网格比**.

可以看出，u_j^n 依赖于 $n-1$ 时间层的 u_j^{n-1},u_{j-1}^{n-1}；而这两个值又依赖于 $n-2$ 时间层的 $u_j^{n-2},u_{j-1}^{n-2},u_{j-2}^{n-2}$；依此类推，可以知道，$u_j^n$ 最终依赖于初始层 $n=0$ 上的下列值(初值条件给出)

$$\varphi_{j-n},\varphi_{j-n+1},\cdots,\varphi_j.$$

因此，称 x 轴上含于区间 $[x_{j-n},x_j]$ 上的网格点为差分解 u_j^n 在点 (x_j,t_n) 的依赖区域，有时也称区间 $[x_{j-n},x_j]$ 为差分解 u_j^n 在点 (x_j,t_n) 的依赖区域. 这依赖区域即是过点 (x_j,t_n) 的两条直线 $x-x_j=\dfrac{h}{\tau}(t-t_n)$ 和 $x=x_j$ 在 x 轴上截出的区间.

如果 $a\dfrac{\tau}{h}>1$，即 $\dfrac{\tau}{h}>\dfrac{1}{a}$，那么微分方程的解 u 在点 (x_j,t_n) 的依赖

区域$[P',Q]$大于差分解 u_j^n 在点(x_j,t_n)的依赖区域$[P,Q]$,见图 12.14.

让网格步长变小,但网格比τ/h保持不变,则依赖区域$[P,Q]$和$[P',Q]$不变.显然,若改变(P',P)内的初值,但$[P,Q]$上的初值不变,则微分方程的初值问题的解$u(x_j,t_n)$可取不同的值,而当 $h\to 0$,$\tau\to 0$ 时(τ/h 不变)差分格式的解 u_j^n 是一串确定的数,它不可能收敛到不同的 $u(x_j,t_n)$,即当$a\tau/h>1$时,差分格式不收敛.由此可以得出差分格式(12.2.10)收敛的必要条件是差分格式的依赖区域包含微分方程初值问题的依赖区域,即

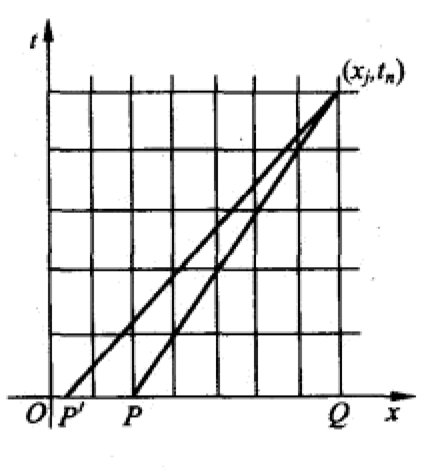


图 12.14

$$a\frac{\tau}{h}\leqslant 1. \qquad (12.2.11)$$

此条件称为**库朗-弗兰特里希斯-勒维**(Courant-Friedrichs-Lewy)**条件**,也称**库朗条件**.可以证明,它也是差分格式(12.2.10)收敛的充分条件.类似的推导可以得出差分格式(12.2.9)是不收敛的.

考虑差分格式(12.2.10)的稳定性问题.设初值 u_j^0 受扰,即$(u+\epsilon)_j^0=u_j^0+\epsilon_j^0$;相应地 u_j^n 也受扰,即$(u+\epsilon)_j^n=u_j^n+\epsilon_j^n$,容易得出 ϵ_j^n 也满足差分方程

$$\frac{\epsilon_j^{n+1}-\epsilon_j^n}{\tau}+a\frac{\epsilon_j^n-\epsilon_{j-1}^n}{h}=0. \qquad (12.2.12)$$

把初始误差 ϵ_j^0 表示为一个简谐波的形式

$$\epsilon_j^0=(e^{i\omega x})_{x=jh}=e^{i\omega jh},$$

其中 ω 为频率参数.要求形如

$$\epsilon_j^n=[G(\omega)]^n e^{i\omega jh} \quad (j=0,\pm 1,\pm 2,\cdots) \qquad (12.2.13)$$

的谐波解，其中 $G=G(\omega)$ 为对应于 ω 的**增长因子**(amplification factor). 将 ϵ_j^n 的表达式(12.2.13)代入式(12.2.12)，整理得

$$\frac{G-1}{\tau}+a\frac{1-e^{-i\omega h}}{h}=0,$$

此方程称为差分方程(12.2.10)的特征方程，它的根即增长因子

$$G=G(\omega)=1-a\lambda(1-e^{-i\omega h}), \qquad (12.2.14)$$

其中 $\lambda=\tau/h$.

对于式(12.2.13)，当 $|G(\omega)|>1$ 时，误差随 n 作指数状增长；$|G(\omega)|\leqslant 1$ 时，误差不增长，由于初始误差可以表示为不同频率 ω 的谐波的叠加，在计算中舍入误差具有随机性，应该认为所有的 ω 的频率组分都是可能出现的，因此数值稳定的条件对所有的实数 ω 都有

$$|G(\omega)|\leqslant 1. \qquad (12.2.15)$$

对于差分格式(12.2.10)，其增长因子为

$$G=1-a\lambda(1-\cos\omega h)-a\lambda i\sin\omega h,$$

$$|G|^2=[1-a\lambda(1-\cos\omega h)]^2+a^2\lambda^2\sin^2\omega h$$

$$=1-4a\lambda(1-a\lambda)\sin^2\frac{\omega h}{2},$$

所以，当 $a\lambda\leqslant 1$ 时有 $|G|\leqslant 1$，即差分格式(12.2.10)在条件 $a\lambda\leqslant 1$ 之下是稳定的，否则不稳定.

差分格式的稳定性判别条件(12.2.15)是具有一般性的. 为了从线性常系数差分方程得到判别稳定性用的特征方程，只要将 u_j^n 以 $G^n e^{i\omega jh}$ 代入相应的差分方程即可.

对于差分格式(12.2.9)，特征方程为

$$\frac{G-1}{\tau}+a\frac{e^{i\omega h}-1}{h}=0,$$

从而得增长因子

$$G=1+a\lambda(1-e^{i\omega h})=1+a\lambda(1-\cos\omega h)-a\lambda i\sin\omega h,$$

$$|G|^2=[1+a\lambda(1-\cos\omega h)]^2+a^2\lambda^2\sin^2\omega h$$

$$= 1 + 4a\lambda(1 + a\lambda)\sin^2 \frac{\omega h}{2}.$$

因此，不可能选择网格比 λ 使 $|G| \leqslant 1$，所以差分格式是不稳定的.

差分格式(12.2.10)在条件 $a\lambda \leqslant 1$ 之下是稳定的，这种在一定条件下稳定的差分格式称为**条件稳定**的差分格式；而像式(12.2.9)那样的差分格式则称为**绝对不稳定**的差分格式. 差分格式(12.2.10)在条件 $a\lambda \leqslant 1$ 之下既稳定也收敛，而差分格式(12.2.9)不但不稳定而且不收敛. 稳定性和收敛性的关系由**拉克斯(Lax)等价定理**给出.

定理 12.2.1 给定一个适定的线性初值问题及其相应的相容的差分格式，则差分格式的收敛性等价于差分格式的稳定性.

所谓适定的初值问题，是指该初值问题的解存在、惟一并连续依赖于初始数据.

12.2.3 对流方程的差分格式

对流方程是最简单的双曲型方程，对其差分格式的研究将有助于求解复杂的双曲型方程和双曲型方程组.

逼近对流方程(12.2.1)的中心差分格式

$$\frac{u_j^{n+1} - u_j^n}{\tau} + a\frac{u_{j+1}^n - u_{j-1}^n}{2h} = 0 \qquad (12.2.16)$$

是很自然的一个差分格式，其截断误差为 $O(\tau + h^2)$. 容易求出其增长因子

$$G(\omega) = 1 - a\lambda \mathrm{i}\sin\omega h,$$

$$|G|^2 = 1 + a^2\lambda^2\sin^2\omega h,$$

因此这是一个绝对不稳定的格式.

把格式(12.2.16)修改为

$$\frac{u_j^{n+1} - \frac{1}{2}(u_{j+1}^n + u_{j-1}^n)}{\tau} + a\frac{u_{j+1}^n - u_{j-1}^n}{2h} = 0, \qquad (12.2.17)$$

称其为**拉克斯-弗里特里希斯**(Lax-Friedrichs)**格式**，容易求出其截断误差 $E=O\left(\frac{h^2}{\tau}\right)+O(\tau+h^2)$. 由于在双曲型方程的计算中，一般采用 $\tau=O(h)$，因而格式(12.2.17)的截断误差化为 $E=O(\tau+h)$. 这是一个一阶精度的格式，其增长因子为

$$G(\omega) = \cos\omega h - \mathrm{i}a\lambda\sin\omega h,$$
$$|G|^2 = 1-(1-a^2\lambda^2)\sin^2\omega h,$$

故格式(12.2.17)的稳定性条件是 $|a|\lambda\leqslant 1$.

为提高差分格式的精度，考虑差分格式

$$\frac{u_j^{n+1}-u_j^{n-1}}{2\tau}+a\frac{u_{j+1}^n-u_{j-1}^n}{2h}=0, \qquad (12.2.18)$$

一般称为**蛙跳格式**(leapfrog scheme). 显然，这是一个二阶精度格式，其稳定性条件为 $|a|\lambda<1$. 要计算 $n+1$ 层上的值 u_j^{n+1}，就必须应用 $n-1,n$ 两个时间层的值 $u_j^{n-1},u_{j+1}^n,u_{j-1}^n$，前后联系到三个时间层次，因此这样的格式称为**三层格式**(three-level scheme). 这种格式在实际计算时，必须保留两个时间层的数据，从而增加计算机的存储量；此外，从第 0 层推进到第 1 层还必须用另外格式来补充，所以使用中有一定的不便.

从中心差分格式(12.2.16)进行修正，还可以得到另一个很有效的格式. 设 $u=u(x,t)$ 是对流方程(12.2.1)的光滑解，把它代入差分格式并进行泰勒级数展开.

$$\frac{u(x_j,t_{n+1})-u(x_j,t_n)}{\tau}+a\frac{u(x_{j+1},t_n)-u(x_{j-1},t_n)}{2h}$$
$$=\left(\frac{\partial u}{\partial t}\right)_j^n+\frac{\tau}{2}\left(\frac{\partial^2 u}{\partial t^2}\right)_j^n+\left(a\frac{\partial u}{\partial x}\right)_j^n+O(\tau^2+h^2).$$

由于 $u=u(x,t)$ 是对流方程的解，所以有

$$\frac{\partial u}{\partial t}=-a\frac{\partial u}{\partial x},$$
$$\frac{\partial^2 u}{\partial t^2}=-a\frac{\partial}{\partial t}\left(\frac{\partial u}{\partial x}\right)=-a\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial t}\right)=a^2\frac{\partial^2 u}{\partial x^2},$$

$$\left(\frac{\tau}{2}\frac{\partial^2 u}{\partial t^2}\right)_j^n = \frac{a^2\tau}{2}\left(\frac{\partial^2 u}{\partial x^2}\right)_j^n$$

$$= \frac{a^2}{2}\frac{\tau}{h^2}(u_{j+1}^n - 2u_j^n + u_{j-1}^n) + O(\tau h^2),$$

由此得到逼近对流方程(12.2.1)的修正中心差分格式

$$\frac{1}{\tau}(u_j^{n+1} - u_j^n) + \frac{a}{2h}(u_{j+1}^n - u_{j-1}^n) - \frac{\tau a^2}{2h^2}(u_{j+1}^n - 2u_j^n + u_{j-1}^n) = 0. \tag{12.2.19}$$

此格式的截断误差 $E=O(\tau^2+h^2)$，所以是二阶精度格式，其稳定性条件是 $|a|\lambda\leqslant 1$. 这是一个二层格式，使用方便，因而受到普遍重视. 格式(12.2.19)也称为**拉克斯-温德罗夫**(Lax-Wendroff)**格式**，其经常使用的还有以下形式：

$$u_j^{n+1} = u_j^n - \frac{1}{2}a\lambda(u_{j+1}^n - u_{j-1}^n) + \frac{1}{2}a^2\lambda^2(u_{j+1}^n - 2u_j^n + u_{j-1}^n),$$

其中 $\lambda=\tau/h$.

当 $a>0$ 时，差分格式(12.2.9)是不稳定的，而格式(12.2.10)在条件 $a\lambda\leqslant 1$ 之下是稳定的；如果设 $a<0$，同样分析可以知道，差分格式(12.2.10)是绝对不稳定的，而格式(12.2.9)在条件 $|a|\lambda\leqslant 1$ 之下是稳定的. 所以，差分格式(12.2.9)和(12.2.10)稳定与否同对流方程(12.2.1)中 a 的符号有关. 由于对流方程(12.2.1)有一族特征线

$$x - at = C \quad (C\text{ 为常数}),$$

当 $a>0$ 时，特征线向右倾斜；当 $a<0$ 时，特征线向左倾斜，因此所构造的差分格式是否稳定，实质上与特征线走向有关. 如果差分格式与微分方程特征线的走向一致，则在 $|a|\lambda\leqslant 1$ 条件下是稳定的；反之，差分格式与微分方程特征线走向不一致，则对任意的网格比都是不稳定的. 沿特征线走向建立的差分格式称为**特征型差**

及初始条件 $u_j^0=\varphi_j(j=0,1,\cdots,J)$，可以显式（不用解方程组）地求解，所以使用方便.

例 12.2.2 用拉克斯-弗里德里希斯格式、迎风(upwind)格式、拉克斯-温德洛夫格式和比姆-沃明格式计算对流方程初值问题

$$\begin{cases}\dfrac{\partial u}{\partial t}+\dfrac{\partial u}{\partial x}=0, & -\infty<x<\infty, t>0 \\ u(x,0)=u_0(x)=\begin{cases}1, & x<0, \\ 0, & x>0,\end{cases}\end{cases} \tag{12.2.22}$$

解 取 $\lambda=\tau/h=0.5$，计算到 $t=0.5$. 图 12.15 表示 $h=0.01$ 的计算结果，图 12.16 表示 $h=0.0025$ 的计算结果，（实线示为解析解，点线为数值解，为比较起见，解析解中增画了一条连接线）.

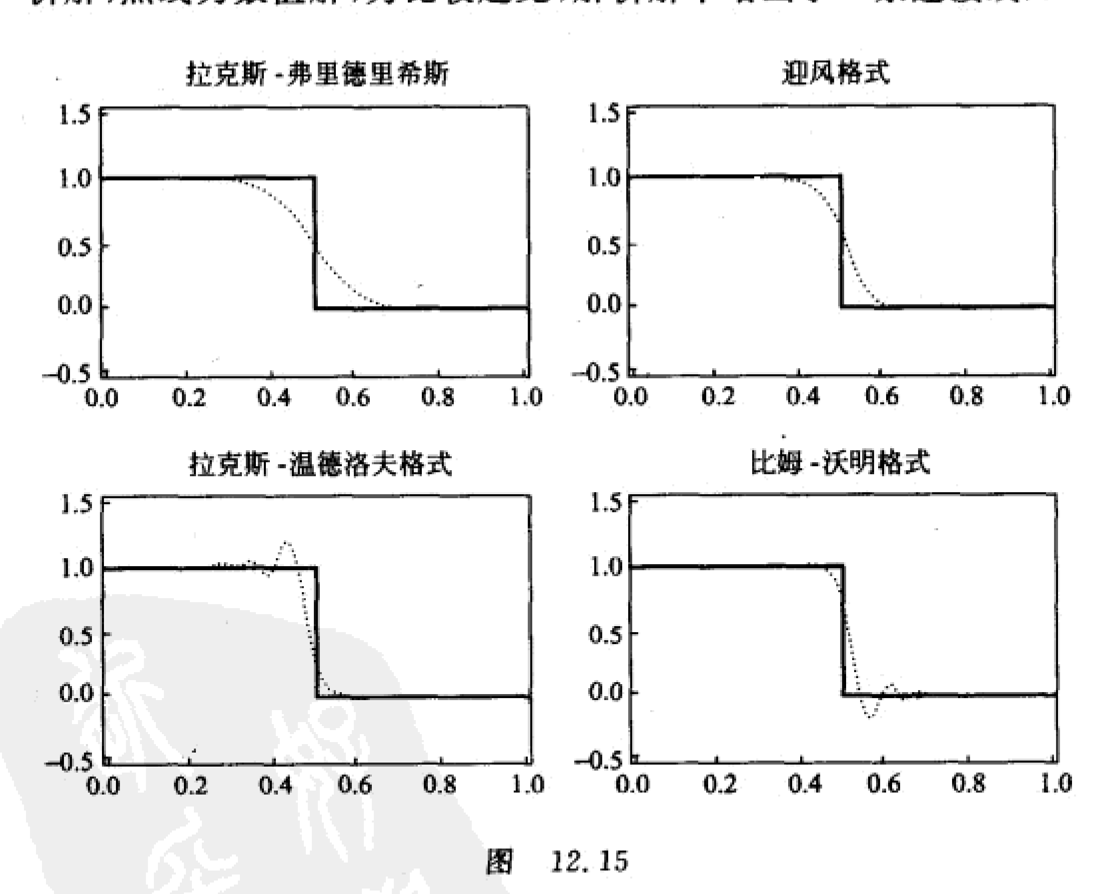


图 12.15

分格式，通常也称**迎风差分格式**(upwind difference scheme).

利用微分方程的特征线以及微分方程的解在特征线上为常数这一事实，还可以构造出**二阶迎风差分格式**. 设 $a>0$，其格式可以写为

$$u_j^{n+1}=u_j^n-a\lambda(u_j^n-u_{j-1}^n)-\frac{a\lambda}{2}(1-a\lambda)(u_j^n-2u_{j-1}^n+u_{j-2}^n), \quad (12.2.20)$$

这个格式的稳定性条件是 $a\lambda\leqslant 2$.

此格式由 R. M. Beam 和 R. F. Warming 于 1976 年提出，所以一般也称其为比姆-沃明(Beam-Warming)格式.

逼近对流方程(12.2.1)的差分格式还可以是

$$\frac{u_j^{n+1}-u_j^n}{\tau}+a\frac{u_j^{n+1}-u_{j-1}^{n+1}}{h}=0 \quad (a>0), \quad (12.2.21)$$

这个格式在第 $n+1$ 时间层上有多于一个点上的未知函数值出现，故称其为**隐式格式**(implicit diffence scheme). 隐式差分格式适用于求解初边值混合问题或具有周期条件的初值问题. 对于稳式格式，一般都要求解代数方程组(但很多双曲型方程的隐式格式则可直接计算). 差分格式(12.2.21)的增长因子为

$$G=\frac{1}{1+a\lambda(1-\mathrm{e}^{-\mathrm{i}\omega h})},$$

所以对任意的 λ 有 $|G|\leqslant 1$，这样的格式称为**绝对稳定的**.

用隐式差分格式(12.2.21)解初边值混合问题.

设求解区域 $\Omega=\{(x,t)\,|\,0\leqslant x\leqslant l, t\geqslant 0\}$，此时网格由直线族

$$x=x_j=jh, \quad h=\frac{l}{J} \quad (j=0,1,\cdots,J),$$

$$t=t_n=n\tau, \quad \tau>0 \quad (n=0,1,\cdots)$$

组成. 设 $a>0$，则 $u_0^{n+1}(n\geqslant 0)$ 给定. 再由方程(12.2.2)，即

$$u_j^{n+1}=\frac{1}{1+a\lambda}u_j^n+\frac{a\lambda}{1+a\lambda}u_{j-1}^{n+1} \quad (j=1,2,\cdots,J-1)$$

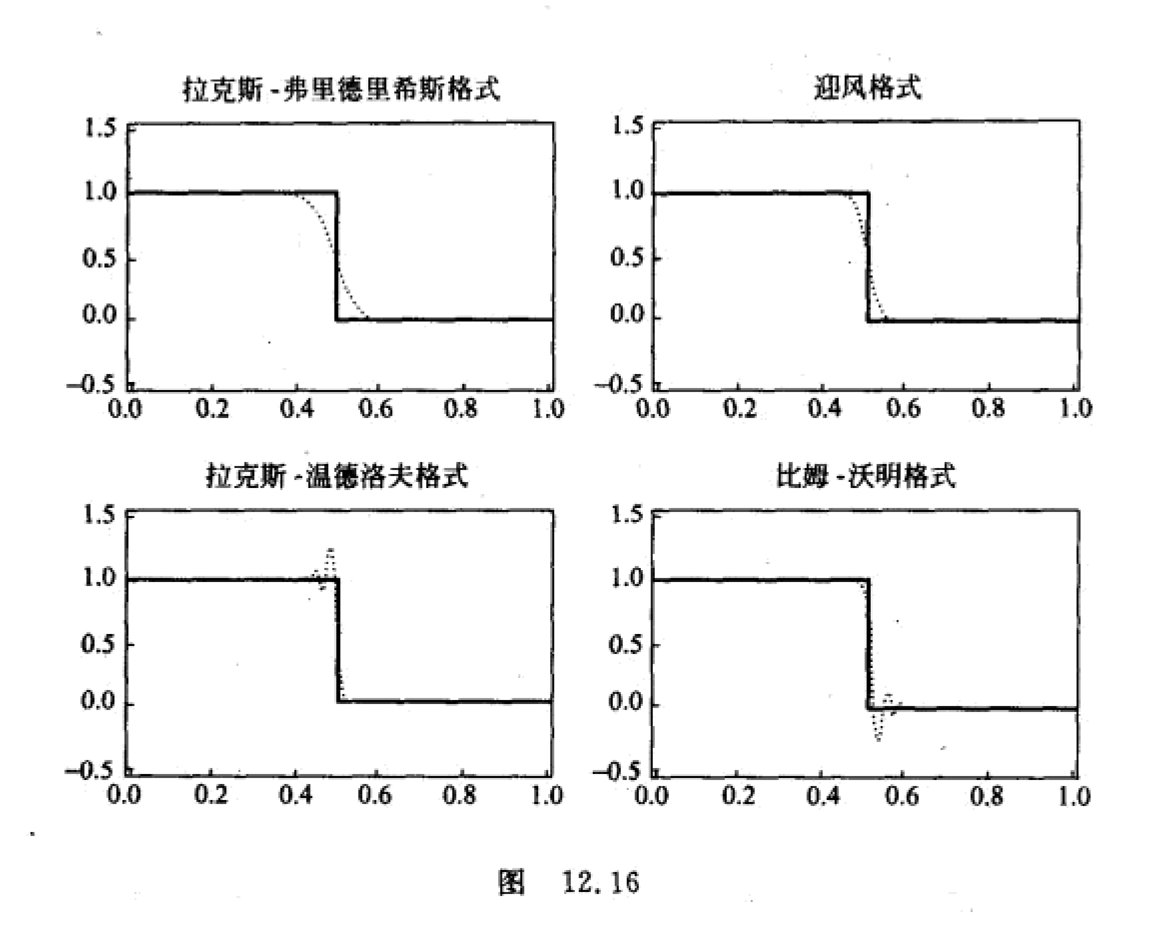


图 12.16

求解对流方程(12.2.1)的常见差分格式如表 12.1 所示.

表 12.1

名 称	格式与截断误差	稳定条件
迎风格式	$\frac{1}{\tau}(u_j^{n+1}-u_j^n)+\frac{a}{h}(u_j^n-u_{j-1}^n)=0$ $E=O(\tau+h)$	$a\lambda\leqslant 1$
右偏显式格式	$\frac{1}{\tau}(u_j^{n+1}-u_j^n)+\frac{a}{h}(u_{j+1}^n-u_j^n)=0$ $E=O(\tau+h)$	绝对不稳定
迎风隐式格式	$\frac{1}{\tau}(u_j^n-u_j^n)+\frac{a}{h}(u_j^{n+1}-u_{j-1}^{n+1})=0$ $E=O(\tau+h)$	绝对稳定

续表

名　称	格式与截断误差	稳定条件
中心显式格式	$\frac{1}{\tau}(u_j^{n+1}-u_j^n)+\frac{a}{2h}(u_{j+1}^n-u_{j-1}^n)=0$ $E=O(\tau+h^2)$	绝对不稳定
中心隐式格式	$\frac{1}{\tau}(u_j^{n+1}-u_j^n)+\frac{a}{2h}(u_{j+1}^{n+1}-u_{j-1}^{n+1})=0$ $E=O(\tau+h^2)$	绝对稳定
平均隐式格式	$\frac{1}{\tau}(u_j^{n+1}-u_j^n)+\frac{1}{2}\Big[\frac{a}{2h}(u_{j+1}^n-u_{j-1}^n)+$ $\frac{a}{2h}(u_{j+1}^{n+1}-u_{j-1}^{n+1})\Big]=0$ $E=O(\tau^2+h^2)$	绝对稳定
拉克斯-弗里德里希斯格式	$\frac{1}{\tau}\Big[u_j^{n+1}-\frac{1}{2}(u_{j+1}^n+u_{j-1}^n)\Big]+\frac{a}{2h}(u_{j+1}^n-$ $u_{j-1}^n)=0$ $E=O(h^2/\tau)+O(\tau+h^2)$	$a\lambda\leqslant 1$
拉克斯-温德洛夫格式	$u_j^{n+1}=u_j^n-\frac{1}{2}a\lambda(u_{j+1}^n-u_{j-1}^n)+\frac{1}{2}a^2\lambda^2$ $(u_{j+1}^n-2u_j^n+u_{j-1}^n)$ $E=O(\tau^2+h^2)$	$a\lambda\leqslant 1$
蛙跳格式	$\frac{u_j^{n+1}-u_j^{n-1}}{2\tau}+a\frac{u_{j+1}^n-u_{j-1}^n}{2h}=0$ $E=O(\tau^2+h^2)$	$a\lambda\leqslant 1$
比姆-沃明格式	$u_j^{n+1}=u_j^n-a\lambda(u_j^n-u_{j-1}^n)-\frac{a\lambda}{2}(1-a\lambda)(u_j^n$ $-2u_{j-1}^n+u_{j-2}^n)$ $E=O(\tau^2+h^2)$	$a\lambda\leqslant 2$
温德洛夫隐式格式	$u_{j+1}^{n+1}=u_j^n+\frac{1-a\lambda}{1+a\lambda}(u_{j+1}^n-u_j^{n+1})$ $E=O(\tau^2+h^2)$	绝对稳定
罗伯茨-维斯隐式格式	$u_j^{n+1}=u_j^n+\frac{a\lambda}{2+a\lambda}(u_{j-1}^{n+1}-u_{j+1}^n)$ $E=O(\tau^2+h)$	绝对稳定

续表

名　称	格式与截断误差	稳定条件
Richtmyer 多步格式	步骤 1 $\dfrac{u_j^{n+\frac{1}{2}}-\dfrac{1}{2}(u_{j+1}^n+u_{j-1}^n)}{\dfrac{\tau}{2}}+a\dfrac{u_{j+1}^n-u_{j-1}^n}{2h}$ 步骤 2 $\dfrac{u_j^{n+1}-u_j^n}{\tau}+a\dfrac{u_{j+1}^{n+\frac{1}{2}}-u_{j-1}^{n+\frac{1}{2}}}{2h}$ $E=O(\tau^2+h^2)$	$a\lambda\leqslant 2$
拉克斯-温德洛夫多步方法	步骤 1 $u_{j+\frac{1}{2}}^{n+\frac{1}{2}}=\dfrac{1}{2}(u_{j+1}^n+u_j^n)-\dfrac{a\lambda}{2}(u_{j+1}^n-u_j^n)$ 步骤 2 $u_j^{n+1}=u_j^n-a\lambda\left(u_{j+\frac{1}{2}}^{n+\frac{1}{2}}-u_{j-\frac{1}{2}}^{n+\frac{1}{2}}\right)$ $E=O(\tau^2+h^2)$	$a\lambda\leqslant 1$
MacCormack 多步方法	$u_j^*=u_j^n-a\lambda(u_{j+1}^n-u_j^n)$ $u_j^{n+1}=\dfrac{1}{2}[(u_j^n+u_j^*)-a\lambda(u_j^*-u_{j-1}^*)]$ $E=O(\tau^2+h^2)$	$a\lambda\leqslant 1$

12.2.4　二维对流方程的差分格式

简单的二维对流扩散方程的初值问题

$$\begin{cases}\dfrac{\partial u}{\partial t}+a\dfrac{\partial u}{\partial x}+b\dfrac{\partial u}{\partial y}=0 \quad (-\infty<x,y<\infty,t>0),\\ u(x,y,0)=f(x,y).\end{cases} \tag{12.2.23}$$

其解为 $u(x,y,t)=f(x-at,y-bt)$.

首先对区域进行离散，$t_n=n\tau,n\geqslant 0,\tau$ 为时间步长. $x_j=jh_x$ $(j=0,\pm 1,\pm 2,\cdots)$；$y_k=kh_y(k=0,\pm 1,\pm 2,\cdots)$，其中 h_x 和 h_y 分别为 x 方向和 y 方向的空间步长. 为简单起见，令 $h_x=h_y=h$，引入记号

$$
\begin{cases}
\delta_{x0} u_{jk} = u_{j+1,k} - u_{j-1,k}, \\
\delta_{x+} u_{jk} = u_{j+1,k} - u_{jk}, \\
\delta_{x-} u_{jk} = u_{jk} - u_{j-1,k}, \\
\delta_{y0} u_{jk} = u_{j,k+1} - u_{j,k-1}, \\
\delta_{y+} u_{jk} = u_{j,k+1} - u_{jk}, \\
\delta_{y-} u_{jk} = u_{jk} - u_{j,k-1}, \\
\delta_x^2 u_{jk} = u_{j+1,k} - 2u_{jk} + u_{j-1,k}, \\
\delta_y^2 u_{jk} = u_{j,k+1} - 2u_{jk} + u_{j,k-1},
\end{cases}
\tag{12.2.24}
$$

方程(12.2.23)的中心格式为

$$
u_{jk}^{n+1} = u_{jk}^n - \frac{a\tau}{2h}\delta_{x0} u_{jk}^n - \frac{b\tau}{2h}\delta_{y0} u_{jk}^n, \tag{12.2.25}
$$

此格式的截断误差为 $O(\tau + h^2 + h^2)$；这个格式是绝对不稳定的，相应的隐式格式为

$$
u_{jk}^{n+1} + \frac{1}{2}\frac{a\tau}{h}\delta_{x0} u_{jk}^{n+1} + \frac{1}{2}\frac{b\tau}{h}\delta_{y0} u_{jk}^{n+1} = u_{jk}^n, \tag{12.2.26}
$$

此格式是无条件稳定的．二维的拉克斯-弗里德里希斯格式是

$$
u_{jk}^{n+1} = \frac{1}{4}(u_{j+1,k}^n + u_{j-1,k}^n + u_{j,k+1}^n + u_{j,k-1}^n) - \frac{1}{2}a\frac{\tau}{h}\delta_{x_0} u_{jk}^n - \frac{1}{2}b\frac{\tau}{h}\delta_{y_0} u_{jk}^n, \tag{12.2.27}
$$

稳定性条件为

$$
(a^2 + b^2)\frac{\tau^2}{h^2} \leqslant \frac{1}{2}.
$$

注意到此稳定性条件较严格，因此采用**近似因子分解方法**.

$$
\begin{cases}
u_{jk}^{n+\frac{1}{2}} = u_{jk}^n - \frac{a\tau}{2h}\delta_{x_0} u_{jk}^n + \frac{1}{2}\left(\frac{a\tau}{h}\right)^2 \delta_x^2 u_{jk}^n, \\
u_{jk}^{n+1} = u_{jk}^{n+\frac{1}{2}} - \frac{b\tau}{2h}\delta_{y_0} u_{jk}^n + \frac{1}{2}\left(\frac{b\tau}{h}\right)^2 \delta_y^2 u_{jk}^{n+\frac{1}{2}}.
\end{cases}
\tag{12.2.28}
$$

交替方向隐式格式有

$$\begin{cases}\left(1+\frac{1}{2}a\frac{\tau}{h}\delta_{x_0}\right)u_{jk}^{n+\frac{1}{2}}=u_{jk}^{n},\\ \left(1+\frac{1}{2}b\frac{\tau}{h}\delta_{y_0}\right)u_{jk}^{n+1}=u_{jk}^{n+\frac{1}{2}}.\end{cases} \tag{12.2.29}$$

此方法也称局部一维方法，其精度为 $O(\tau+h^2+h^2)$，并是无条件稳定性，常见的二维对流方程格式见表 12.2.

表 12.2

名 称	格式与截断误差	稳定性条件
中心格式	$u_{jk}^{n+1}=u_{jk}^{n}-\frac{a}{2}\frac{\tau}{h}\delta_{x_0}u_{jk}^{n}-\frac{b}{2}\frac{\tau}{h}\delta_{y_0}u_{jk}^{n}$ $E=O(\tau+h^2+h^2)$	绝对不稳定
拉克斯-弗里德里希斯格式	$u_{jk}^{n+1}=\frac{1}{4}(u_{j+1,k}^{n}+u_{j-1,k}^{n}+u_{j,k+1}^{n}+u_{j,k-1}^{n})-$ $\frac{a}{2}\frac{\tau}{h}\delta_{x_0}u_{jk}^{n}-\frac{b}{2}\frac{\tau}{h}\delta_{y_0}u_{jk}^{n}$	$\left(a\frac{\tau}{h}\right)^2+\left(b\frac{\tau}{h}\right)^2\leqslant\frac{1}{2}$
近似分裂格式（一维拉克斯-温德洛夫格式）	$u_{jk}^{n+\frac{1}{2}}=u_{jk}^{n}-\frac{a}{2}\frac{\tau}{h}\delta_{x_0}u_{jk}^{n}+\frac{1}{2}\left(a\frac{\tau}{h}\right)^2\delta_x^2u_{jk}^{n}$ $u_{jk}^{n+1}=u_{jk}^{n+\frac{1}{2}}-\frac{a}{2}\frac{\tau}{h}\delta_{y_0}u_{jk}^{n+\frac{1}{2}}+$ $\frac{1}{2}\left(b\frac{\tau}{h}\right)^2\delta_y^2u_{jk}^{n+\frac{1}{2}}$ $E=O(\tau^2+h^2+h^2)$	$\max\left\{\lvert a\rvert\frac{\tau}{h},\lvert b\rvert\frac{\tau}{h}\right\}\leqslant 1$
局部一维格式	$\left(1+\frac{a}{2}\frac{\tau}{h}\delta_{x_0}\right)u_{jk}^{n+\frac{1}{2}}=u_{jk}^{n}$ $\left(1+\frac{b}{2}\frac{\tau}{h}\delta_{y_0}\right)u_{jk}^{n+1}=u_{jk}^{n+\frac{1}{2}}$ $E=O(\tau+h^2+h^2)$	无条件稳定
克拉克-尼科尔森格式	$\left(1+\frac{a}{4}\frac{\tau}{h}\delta_{x_0}+\frac{b}{4}\frac{\tau}{h}\delta_{y_0}\right)u_{jk}^{n+1}$ $=\left(1-\frac{a}{4}\frac{\tau}{h}\delta_{x_0}-\frac{b}{4}\frac{\tau}{h}\delta_{y_0}\right)u_{jk}^{n}$ $E=O(\tau^2+h^2+h^2)$	无条件稳定

续表

名　称	格式与截断误差	稳定性条件
比姆-沃明格式	$\left(1+\frac{a}{4}\frac{\tau}{h}\delta_{x_0}\right)u_{jk}^{*}=\left(1-\frac{a}{4}\frac{\tau}{h}\delta_{x_0}\right)\left(1-\frac{b}{4}\frac{\tau}{h}\delta_{y_0}\right)u_{jk}^{n}$ $\left(1+\frac{b}{4}\frac{\tau}{h}\delta_{y_0}\right)u_{jk}^{n+1}=u_{jk}^{*}$ $E=O(\tau^2+h^2+h^2)$	无条件稳定
Delta 公式	$\left(1+\frac{a}{4}\frac{\tau}{h}\delta_{x_0}\right)\Delta u_{jk}^{*}=-\left(\frac{a}{2}\frac{\tau}{h}\delta_{x_0}-\frac{b}{2}\frac{\tau}{h}\delta_{y_0}\right)u_{jk}^{n}$ $\left(1+\frac{b}{4}\frac{\tau}{h}\delta_{y_0}\right)\Delta u_{jk}=\Delta u_{jk}^{*}$ $\Delta u_{jk}=u_{jk}^{n+1}-u_{jk}^{n}$ $E=O(\tau^2+h^2+h^2)$	无条件稳定

12.2.5 波动方程的差分格式

波动方程(12.2.4)的差分格式可以直接由差商代替微商而得到

$$\frac{u_j^{n+1}-2u_j^n+u_j^{n-1}}{\tau^2}=a^2\frac{u_{j+1}^n-2u_j^n+u_{j-1}^n}{h^2}.\quad(12.2.30)$$

这是一个二阶精度的显式格式,其稳定性条件是 $a\lambda<1$,其中 $\lambda=\tau/h$. 隐式格式的形式很多,差分格式

$$\frac{u_j^{n+1}-2u_j^n+u_j^{n-1}}{\tau^2}=a^2\frac{u_{j+1}^{n+1}-2u_j^{n+1}+u_{j-1}^{n+1}}{h^2}\quad(12.2.31)$$

是一个隐式格式,其截断误差为 $O(\tau+h^2)$,这是一个绝对稳定的差分格式. 更一般的差分格式是冯·诺伊曼(von Neumann)格式

$$\frac{u_j^{n+1}-2u_j^n+u_j^{n-1}}{\tau^2}=a^2\left\{\theta\frac{u_{j+1}^{n+1}-2u_j^{n+1}+u_{j-1}^{n+1}}{h^2}+\right.$$

$$(1-2\theta)\frac{u_{j+1}^{n}-2u_{j}^{n}+u_{j-1}^{n}}{h^{2}}+\theta\frac{u_{j+1}^{n-1}-2u_{j}^{n-1}+u_{j-1}^{n-1}}{h^{2}}\Bigg\},$$

(12.2.32)

其中 $0\leqslant\theta\leqslant1$. 利用泰勒级数展开，可以直接验证对任意的 $\theta\in[0,1]$，截断误差都是 $O(\tau^2+h^2)$. 此格式的稳定性条件是：如果 $\frac{1}{4}\leqslant\theta\leqslant1$，则差分格式绝对稳定；如果 $0\leqslant\theta<\frac{1}{4}$，则差分格式的稳定性条件为 $0<a\lambda<\frac{1}{\sqrt{1-4\theta}}$. 特别地，当 $\theta=0$ 时，差分格式(12.2.32)化为显式格式.

格式(12.2.32)的另外两个重要的特殊情形是 $\theta=\frac{1}{2}$ 和 $\theta=\frac{1}{4}$. 当 $\theta=\frac{1}{2}$ 时，差分格式化为

$$\frac{u_{j}^{n+1}-2u_{j}^{n}+u_{j}^{n-1}}{\tau^{2}}$$
$$=a^{2}\ \frac{1}{2}\left[\frac{u_{j+1}^{n+1}-2u_{j}^{n+1}+u_{j-1}^{n+1}}{h^{2}}+\frac{u_{j+1}^{n-1}-2u_{j}^{n-1}+u_{j-1}^{n-1}}{h^{2}}\right],$$

此格式称为**平均隐式格式**，由于 $\theta\geqslant\frac{1}{4}$，所以是稳定的. $\theta=\frac{1}{4}$ 在实际应用中是很受重视的一个差分格式.

关于初始条件的差分近似，对于第一个初始条件 $u(x,0)=\varphi(x)(-\infty<x<\infty)$，可以用直接转移的方法，得到

$$u_{j}^{0}=\varphi(jh)=\varphi_{j}\quad(j=0,\pm1,\pm2,\cdots).\qquad(12.2.33)$$

对于第二个初始条件 $\frac{\partial u(x,0)}{\partial t}=\psi(x)(-\infty<x<\infty)$，一种方法是

$$\frac{u_{j}^{1}-u_{j}^{0}}{\tau}=\psi(jh)=\psi_{j}\quad(j=0,\pm1,\pm2,\cdots).$$

(12.2.34)

容易看出,式(12.2.34)的截断误差为$O(\tau)$,如果采用冯·诺伊曼差分格式$\left(包括\ \theta=0,\frac{1}{4},\frac{1}{2}\right)$进行计算,那么初始条件的差分近似与方程的差分近似不适应.为提高式(12.2.34)的截断误差可采用另一种方法,即

$$\frac{u_j^1-u_j^{-1}}{2\tau}=\psi_j \quad (j=0,\pm 1,\pm 2,\cdots). \qquad (12.2.35)$$

这种方法的截断误差为$O(\tau^2)$.但是,因为它又引进了新的未知数u_j^{-1},所以由式(12.2.35)不能确定u_j^1.为确定u_j^1,应把式(12.2.35)和在$(x_j,0)$上的差分方程联立求出u_j^1.设采用显式格式进行计算,那么在$(x_j,0)$上有

$$\frac{u_j^1-2u_j^0+u_j^{-1}}{\tau^2}=a^2\,\frac{u_{j+1}^0-2u_j^0+u_{j-1}^0}{h^2},$$

此式与式(12.2.35)联立,可以消去u_j^{-1}并得到

$$u_j^1=\frac{a^2\lambda^2}{2}(\varphi_{j-1}+\varphi_{j+1})+(1-a^2\lambda^2)\varphi_j+\tau\psi_j, \qquad (12.2.36)$$

其中$\lambda=\tau/h$.

有了初始条件的离散,利用式(12.2.33)、式(12.2.34)或式(12.2.36)可以算出初始层($n=0$)及第一层($n=1$)各网格点上的值,然后再利用差分格式逐层算出任意网格点上的值.

关于波动方程的初边值混合问题.由求解区域$\Omega=\{(x,t)\mid 0\leqslant x\leqslant l,t\geqslant 0\}$,因此网格由直线

$$x=x_j=jh,h=\frac{l}{J} \quad (j=0,1,2,\cdots,J),$$

$$t=t_n=n\tau \quad (\tau>0,n\geqslant 0)$$

所构成.对于边界条件的离散,有以下方法:

(1) 第一类边界条件采用直接转移,

$$u_0^n=\Phi_0^n=\Phi_0(n\tau),\quad u_J^n=\Phi_1^n=\Phi_1(n\tau). \qquad (12.2.37)$$

(2) 第三类边界条件采用两种办法. 第一种办法是直接用差商代替微商,即

$$\begin{cases}\dfrac{u_1^n - u_0^n}{h} + \eta_0^n u_0^n = F_0^n, \\ \dfrac{u_J^n - u_{J-1}^n}{h} + \eta_1^n u_J^n = F_1^n,\end{cases} \tag{12.2.38}$$

其中

$$\eta_0^n = \eta_0(n\tau), \eta_1^n = \eta_1(n\tau), F_0^n = F_0(n\tau), F_1 = F_1(n\tau).$$

其截断误差为 $O(h)$. 第二种办法是将网格平移半个步长,即在网格

$$\begin{cases}x_j = \left(j + \dfrac{1}{2}\right)h, h = \dfrac{l}{J} \quad (j = -1, 0, 1, \cdots, J), \\ t_n = n\tau \quad (n \geqslant 0)\end{cases} \tag{12.2.39}$$

上讨论差分近似. 此情况下第三边界条件的差分近似是

$$\begin{cases}\dfrac{u_0^n - u_{-1}^n}{h} + \eta_0^n \dfrac{u_0^n + u_{-1}^n}{2} = F_0^n, \\ \dfrac{u_J^n - u_{J-1}^n}{h} + \eta_1^n \dfrac{u_J^n + u_{J-1}^n}{2} = F_1^n,\end{cases} \tag{12.2.40}$$

其截断误差为 $O(h^2)$. 利用这一差分近似时,在算出所有的 u_j^n 在内部节点上的值以后,应再用插值法求出 $x=0$ 和 $x=l$ 上 u 的值.

12.3 抛物型方程的差分方法

12.3.1 典型问题

抛物型方程(parabolic equations)的最简单方程是**扩散方程**(diffusion equation),即

$$\frac{\partial u}{\partial t} = b\frac{\partial^2 u}{\partial x^2} \quad (-\infty < x < \infty, t \geqslant 0), \tag{12.3.1}$$

其中 $b>0$. 此方程也称热传导方程,如果给定初始条件

$$u(x,0) = \varphi(x) \quad (-\infty < x < \infty), \tag{12.3.2}$$

就构成了初值问题. 这个初值问题的解可以表示为

$$u(x,t)=\frac{1}{\sqrt{4\pi bt}}\int_{-\infty}^{\infty}\exp\left[-\frac{(x-\xi)^2}{4bt}\right]\varphi(\xi)\mathrm{d}\xi$$

$$(-\infty<x<\infty,t>0). \qquad (12.3.3)$$

对于扩散方程(12.3.1)还有初边值混合问题,即在区域 $\Omega=\{(x,t)|0\leqslant x\leqslant l,t\geqslant 0\}$ 内考虑扩散方程的求解,除了给出在 Ω 内的方程(12.3.1)和初始条件(12.3.2)之外,还要给出边界条件. 边界条件提法如下:

(1) 第一类边界条件

$$\begin{cases}u(0,t)=\mu_1(t), & t\geqslant 0,\\ u(l,t)=\mu_2(t), & t\geqslant 0.\end{cases} \qquad (12.3.4)$$

(2) 第三类边界条件

$$\begin{cases}\left.\left(\dfrac{\partial u}{\partial x}-\lambda_1(t)u\right)\right|_{x=0}=v_1(t), & t\geqslant 0,\\ \left.\left(\dfrac{\partial u}{\partial x}+\lambda_2(t)u\right)\right|_{x=l}=v_2(t), & t\geqslant 0,\end{cases} \qquad (12.3.5)$$

其中 $\lambda_i(t)\geqslant 0(i=1,2)$. 如果在式(12.3.5)中, λ_1,λ_2 为零,则称为**第二类边界条件**.

抛物型方程的另一典型例子是**对流扩散方程**(convection diffusion equation)

$$\frac{\partial u}{\partial t}+a\frac{\partial u}{\partial x}=b\frac{\partial^2 u}{\partial x^2}\quad(-\infty<x<\infty,t\geqslant 0), \qquad (12.3.6)$$

其中 $b>0$. 这个方程可以看成是对流方程和扩散方程的耦合,当 b 很小时,方程反映对流方程的特征,当 a 很小时,方程反映扩散方程的特征. 如果仍取式(12.3.2)作为初始条件,则构成对流扩散方程的初值问题,这个初值问题的解可以表示为

$$u(x,t)=\frac{1}{\sqrt{4\pi bt}}\int_{-\infty}^{\infty}\exp\left[-\frac{(x-\xi-at)^2}{4bt}\right]\varphi(\xi)\mathrm{d}\xi. \qquad (12.3.7)$$

最简单的二维扩散方程的初边值问题为

$$\begin{cases} \dfrac{\partial u}{\partial t} = b\left(\dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2}\right), \quad 0 < x < l, \quad 0 < y < l, \quad t > 0, \quad b > 0, \\ u(x,y,0) = \varphi(x,y), \quad 0 \leqslant x, \quad y \leqslant l, \\ u(0,y,t) = \mu_1(y,t), u(l,y,t) = \mu_2(y,t), \\ \qquad\qquad 0 \leqslant y \leqslant l, \quad t \geqslant 0, \\ u(x,0,t) = v_1(x,t), u(x,l,t) = v_2(x,t), \\ \qquad\qquad 0 \leqslant x \leqslant l, \quad t \geqslant 0. \end{cases} \tag{12.3.8}$$

12.3.2 扩散方程的差分格式

初值问题的网格剖分如下：令 h 和 τ 分别是 x 轴方向和 t 轴正方向的步长，$x_j = jh(j=0,\pm1,\cdots)$，$t_n = n\tau(n=0,1,\cdots)$，两组平行线构成了 Oxt 上半平面的网格. 利用差商替代微商的办法可以获得一系列逼近扩散方程(12.3.1)的差分格式.

(1) 显式格式

$$\frac{u_j^{n+1} - u_j^n}{\tau} = b\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{h^2}, \tag{12.3.9}$$

其截断误差为 $O(\tau + h^2)$，增长因子为

$$G = 1 - 4b\lambda\sin^2\frac{\omega h}{2},$$

其中 $\lambda = \tau/h^2$，称 λ 为**网格比**. 但是，此处与双曲型方程的差分格式中的网格比含义不同. 由增长因子 G，可以得格式(12.3.9)的稳定性条件为

$$b\lambda \leqslant \frac{1}{2},$$

即

$$b\frac{\tau}{h^2} \leqslant \frac{1}{2}.$$

(2) 隐式格式

$$\frac{u_j^{n+1}-u_j^n}{\tau}=b\frac{u_{j+1}^{n+1}-2u_j^{n+1}+u_{j-1}^{n+1}}{h^2}, \qquad (12.3.10)$$

其截断误差为 $O(\tau+h^2)$，增长因子为

$$G=\frac{1}{1+4b\lambda\sin^2\frac{\omega h}{2}}.$$

由此可知，此格式对任何网格比都是稳定的.

(3) 克兰克-尼科尔松格式

$$\frac{u_j^{n+1}-u_j^n}{\tau}=b\frac{1}{2}\left\{\frac{u_{j+1}^n-2u_j^n+u_{j-1}^n}{h^2}+\frac{u_{j+1}^{n+1}-2u_j^{n+1}+u_{j-1}^{n+1}}{h^2}\right\}. \qquad (12.3.11)$$

这是一个绝对稳定的隐式格式，特别是它的截断误差为 $O(\tau^2+h^2)$，即为二阶精度格式. 这个格式称为**克兰克-尼科尔森**(Crank-Nicolson)**格式**，简记为 C-N 格式.

(4) 里查森格式

$$\frac{u_j^{n+1}-u_j^{n-1}}{2\tau}=b\frac{u_{j+1}^n-2u_j^n+u_{j-1}^n}{h^2}. \qquad (12.3.12)$$

这是一个二阶精度的三层格式，其增长因子为

$$G_{1,2}=-4b\lambda\sin^2\frac{\omega h}{2}\pm\sqrt{1+\left(4b\lambda\sin^2\frac{\omega h}{2}\right)^2}.$$

因此**里查森**(Richardson)**格式**是一个绝对不稳定的格式.

(5) 杜福特-弗兰克尔格式

$$\frac{u_j^{n+1}-u_j^{n-1}}{2\tau}=b\frac{u_{j+1}^n-(u_j^{n+1}+u_j^{n-1})+u_{j-1}^n}{h^2}. \qquad (12.3.13)$$

杜福特-弗兰克尔(Du Fort-Frankel)**格式**是里查森格式的一种修正，是绝对稳定的差分格式. 其截断误差

$$E=b\left(\frac{\tau}{h}\right)^2\left(\frac{\partial^2 u}{\partial t^2}\right)_j^n+O(\tau^2+h^2)+O\left(\frac{\tau^4}{h^2}\right).$$

如果取 $\tau=O(h)$，则当 $\tau,h\to 0$ 时 E 并不趋于 0，所以差分格式

(12.3.13)与微分方程(12.3.1)是不相容的.只有当 τ,h 趋于 0 时 $\frac{\tau}{h}$ 也趋于 0,差分格式(12.3.13)才能相容于扩散方程(12.3.1).

逼近扩散方程(12.3.1)的常见差分格式见表 12.3.

表 12.3

名　称	差分格式与截断误差	稳定条件
显式格式	$\frac{1}{\tau}(u_j^{n+1}-u_j^n)=b\frac{1}{h^2}(u_{j+1}^n-2u_j^n+u_{j-1}^n)$ $E=O(\tau+h^2)$	$b\lambda\leqslant\frac{1}{2}$
隐式格式	$\frac{1}{\tau}(u_j^{n+1}-u_j^n)=b\frac{1}{h^2}(u_{j+1}^{n+1}-2u_j^{n+1}+u_{j-1}^{n+1})$ $E=O(\tau+h^2)$	绝对稳定
里查森格式	$\frac{1}{2\tau}(u_j^{n+1}-u_j^{n-1})=b\frac{1}{h^2}(u_{j+1}^n-2u_j^n+u_{j-1}^n)$ $E=O(\tau^2+h^2)$	绝对不稳定
克拉克-尼科尔森格式	$\frac{1}{\tau}(u_j^{n+1}-u_j^n)=b\frac{1}{2h^2}\left[(u_{j+1}^{n+1}-2u_j^{n+1}+u_{j-1}^{n+1})+(u_{j+1}^n-2u_j^n+u_{j-1}^n)\right]$ $E=O(\tau^2+h^2)$	绝对稳定
加权隐式格式	$\frac{1}{\tau}(u_j^{n+1}-u_j^n)=b\frac{1}{h^2}\left[\theta(u_{j+1}^{n+1}-2u_j^{n+1}+u_{j-1}^{n+1})+(1-\theta)(u_{j+1}^n-2u_j^n+u_{j-1}^n)\right]$ $E=b\left(\frac{1}{2}-\theta\right)\cdot O(\tau)+O(\tau^2+h^2)$	$\frac{1}{2}\leqslant\theta\leqslant1$, 绝对稳定 $0\leqslant\theta<\frac{1}{2}$, $b\lambda\leqslant\frac{1}{2(1-2\theta)}$
	$\theta\geqslant0$: $(1+\theta)\frac{u_j^{n+1}-u_j^n}{\tau}-\theta\frac{u_j^n-u_j^{n-1}}{\tau}$ $=b\frac{1}{h^2}(u_{j+1}^{n+1}-2u_j^{n+1}+u_{j-1}^{n+1})$ $E=O(\tau+h^2)$	绝对稳定

续表

名　称	差分格式与截断误差	稳定条件
加权隐式格式	$\theta=\frac{1}{2}$： $\frac{3}{2}\frac{u_j^{n+1}-u_j^n}{\tau}-\frac{1}{2}\frac{u_j^n-u_j^{n-1}}{\tau}$ $=b\frac{1}{h^2}(u_{j+1}^{n+1}-2u_j^{n+1}+u_{-1}^{j+1})$ $E=O(\tau^2+h^2)$	绝对稳定
	$\frac{1}{12\tau}(u_{j+1}^{n+1}-u_{j+1}^n)+\frac{5}{6\tau}(u_j^{n+1}-u_j^n)+$ $\frac{1}{12\tau}(u_{j-1}^{n+1}-u_{j-1}^n)$ $=b\frac{1}{2}\left[\frac{1}{h^2}(u_{j+1}^{n+1}-2u_j^{n+1}+u_{j-1}^{n+1})+\right.$ $\left.\frac{1}{h^2}(u_{j+1}^n-2u_j^n+u_{j-1}^n)\right]$ $E=O(\tau^2+h^4)$	绝对稳定
	$\frac{1}{12\tau}\left(\frac{3}{2}u_{j+1}^{n+1}-2u_{j+1}^n+\frac{1}{2}u_{j+1}^{n-1}\right)+$ $\frac{5}{6\tau}\left(\frac{3}{2}u_j^{n+1}-2u_j^n+\frac{1}{2}u_j^{n-1}\right)+$ $\frac{1}{12\tau}\left(\frac{3}{2}u_{j-1}^{n+1}-2u_{j-1}^n+\frac{1}{2}u_{j-1}^{n-1}\right)$ $=b\frac{1}{h^2}(u_{j+1}^{n+1}-2u_j^{n+1}+u_{j-1}^{n+1})$ $E=O(\tau^2+h^4)$	绝对稳定
杜福特-弗兰克尔格式	$\frac{u_j^{n+1}-u_j^{n-1}}{2\tau}=b\frac{u_{j+1}^n-(u_j^{n-1}+u_j^{n+1})+u_{j-1}^n}{h^2}$ $E=O(\tau^2+h^2)+O\left(\frac{\tau^2}{h^2}\right)$	绝对稳定

续表

名　称	差分格式与截断误差	稳定条件
跳点格式	$\frac{1}{\tau}(u_j^{n+1}-u_j^n)-\frac{b}{h^2}(u_{j+1}^n-2u_j^n+u_{j-1}^n)=0$ $n+1+j=$偶数 $\frac{1}{\tau}(u_j^{n+1}-u_j^n)-\frac{b}{h^2}(u_{j+1}^{n+1}-2u_j^{n+1}+u_{j-1}^{n+1})=0$ $n+1+j=$奇数 $E=O(\tau^2+h^2)+O\left(\frac{\tau^2}{h^2}\right)$	绝对稳定
Saul'ev 格式	$u_j^{n+1}=u_j^n+b\frac{\tau}{h^2}(u_{j+1}^n-u_j^n-u_j^{n+1}+u_{j-1}^{n+1})$ $u_j^{n+2}=u_j^{n+1}+b\frac{\tau}{h^2}(u_{j+1}^{n+2}-u_j^{n+2}-u_j^{n+1}+u_{j-1}^{n+1})$ $E=O(\tau^2+h^2)+O\left(\frac{\tau}{h}\right)$	绝对稳定

对于扩散方程初值问题(12.3.1)和(12.3.2)的求解,还必须有初始条件(12.3.2)的离散,这可以用直接转移法得到

$$u_j^0 = \varphi(jh) = \varphi_j, \qquad (12.3.14)$$

再由式(12.3.1)的差分方程及式(12.3.14)进行逐层计算.

关于初边值混合问题的求解,网格不同于初值问题.此时设

$$x_j = jh, \quad h = \frac{l}{J}, \quad (j = 0,1,\cdots,J,)$$

$$t_n = n\tau, \quad (n = 0,1,2,\cdots,\tau\text{ 为时间步长}).$$

对于第一类边界条件,可以用直接转移法,条件(12.3.4)的离散为

$$u_0^n = \mu_1(n\tau), u_J^n = \mu_2(n\tau), \quad n \geqslant 0. \qquad (12.3.15)$$

由于第三类边界条件中含有导数,所以不能用直接转移的办法,通常采用下面两种方法(类同于对波动方程的处理)来处理:

(1) 第一种方法

$$\begin{cases}\dfrac{u_1^n - u_0^n}{h} - \lambda_1(n\tau)u_0^n = v_1(n\tau),\\[2mm] \dfrac{u_J^n - u_{J-1}^n}{h} + \lambda_2(n\tau)u_J^n = v_2(n\tau).\end{cases} \tag{12.3.16}$$

(2) 第二种方法

将网格平移半个步长，即在网格

$$\begin{cases}x_j = \left(j+\dfrac{1}{2}\right)h, h = \dfrac{l}{J} \quad (j=-1,0,1,\cdots,J),\\ t_n = n\tau \quad (n=0,1,2,\cdots)\end{cases}$$

上讨论差分近似. 在此情况下，第三类边界条件的差分近似为

$$\begin{cases}\dfrac{u_0^n - u_{-1}^n}{h} - \lambda_1(n\tau)\dfrac{u_0^n + u_{-1}^n}{2} = v_1(n\tau),\\[2mm] \dfrac{u_J^n - u_{J-1}^n}{h} + \lambda_2(n\tau)\dfrac{u_J^n + u_{J-1}^n}{2} = v_2(n\tau).\end{cases} \tag{12.3.17}$$

在计算出所有的 u_j^n 在内部节点上的值之后，再用插值线求出 $x=0$ 和 $x=l$ 上的值.

例 12.3.1 扩散方程的初边值混合问题的求解.

$$\begin{cases}\dfrac{\partial u}{\partial t} = \dfrac{\partial^2 u}{\partial x^2}, & 0 > x > 1, t > 0,\\ u(0,t) = u(1,t) = 0, & t > 0,\\ u(x,0) = \sin(\pi x), & 0 \leqslant x \leqslant 1.\end{cases}$$

分别用隐式格式、克拉克-尼科尔森格式和显式格式解之，并比较其结果.

解 取空间步长 $h=0.1$，时间步长 $\tau=0.01$，网格比 $\lambda=\frac{\tau}{h^2}=1$.

(1) 隐式格式

$$\begin{cases}\dfrac{u_j^{n+1} - u_j^n}{\tau} = \dfrac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{h^2} \quad (j=1,2,\cdots,9),\\ u_0^{n+1} = u_{10}^{n+1} = 0,\\ u_j^0 = \sin(\pi x_j) \quad (j=0,1,2,\cdots,9,10).\end{cases}$$

考虑到$\lambda=\tau/h^2=1$,可以把差分格式写为

$$\begin{cases}-u_{j-1}^{n+1}+3u_j^{n+1}-u_{j+1}^{n+1}=u_j^n \quad (j=1,2,\cdots,9),\\ u_0^{n+1}=u_{10}^{n+1}=0,\\ u_j^0=\sin(\pi x_j) \quad (j=0,1,\cdots,9,10).\end{cases}$$

写成矩阵形式,有

$$\begin{pmatrix} 3 & -1 & & & \\ -1 & 3 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 3 & -1 \\ & & & -1 & 3 \end{pmatrix}\begin{pmatrix} u_1^{n+1} \\ u_2^{n+1} \\ \vdots \\ u_9^{n+1} \end{pmatrix}=\begin{pmatrix} u_1^n \\ u_2^n \\ \vdots \\ u_9^n \end{pmatrix}.$$

这是以对角占优的三对角线矩阵为系数矩阵的线性代数方程组,可以用追赶法解之.重复计算,可以得到$t=0.5$的一组解,其结果见表 12.4.

表 12.4

x_j	解析解 $u(x_j,0.5)$	全隐式 u_j^{50}	显式 u_j^{50}	C-N 格式 u_j^{50}
0	0	0	0	0
0.1	0.00222241	0.00289802	8.19876×10^7	0.00230512
0.2	0.00422728	0.00551236	-1.55719×10^8	0.00438461
0.3	0.00581836	0.00758711	2.13833×10^8	0.00603489
0.4	0.00683989	0.00891918	-2.50642×10^8	0.0070944
0.5	0.00719188	0.00937818	2.62685×10^8	0.00745954
0.6	0.00683989	0.00891918	-2.49015×10^8	0.0070944
0.7	0.00581836	0.00758711	2.11200×10^8	0.00603489
0.8	0.00422728	0.00551236	-1.53086×10^8	0.00438461
0.9	0.00222241	0.00289802	8.03604×10^7	0.00230512
1.0	0	0	0	0

(2) 克拉克-尼科尔森格式

$$\frac{u_j^{n+1}-u_j^n}{\tau}=\frac{1}{2}\Big(\frac{u_{j+1}^n-2u_j^n+u_{j-1}^n}{h^2}+$$

$$\frac{u_{j+1}^{n+1}-2u_j^{n+1}+u_{j-1}^{n+1}}{h^2}\Big)\quad(j=1,2,\cdots,9).$$

由于$\lambda=\tau/h^2=1$,所以上述方程组化为

$$-u_{j-1}^{n+1}+4u_j^{n+1}-u_{j+1}^{n+1}=u_{j+1}^n+u_{j-1}^n\quad(j=1,2,\cdots,9),$$

此仍是以对角占优的三对角线矩阵为系数的线性代数方程组,可用追赶法解之.重复计算,可以得到$t=0.5$的一组解,其结果见表12.4.

(3) 显式格式

$$u_j^{n+1}=u_{j-1}^n-u_j^n+u_{j+1}^n\quad(j=1,2,\cdots,n).$$

由初始条件开始,直接计算可得到$t=0.5$的一组解,其结果见表12.4.

本题的解析解为

$$u(x,t)=\mathrm{e}^{-\pi^2t}\sin(\pi x).$$

为比较起见,其在网格点的值也列在表12.4中.

由表12.4可以看出,克拉克-尼科尔森格式的结果较为精确,而显式格式的结果已面貌皆非了.这是因为$\lambda=1$,格式已不稳定,所以计算不出正确的数值结果.如果对于显式格式取$\lambda=0.05$即取$\tau=0.0005$,此时可得出与克拉克-尼科尔森相近的结果.所以,在扩散方程的求解中,克拉克-尼科尔森格式是十分可取的,而显式格式则一般不宜采用.

12.3.3 对流扩散方程的差分方法

把对流方程的差分格式和扩散方程的差分格式结合起来,容易构造出对流扩散方程的差分格式.

逼近对流扩散方程(12.3.6)的中心显式格式为

$$\frac{u_j^{n+1}-u_j^n}{\tau}+a\frac{u_{j+1}^n-u_{j-1}^n}{2h}=b\frac{u_{j+1}^n-2u_j^n+u_{j-1}^n}{h^2},$$

(12.3.18)

其截断误差为$O(\tau+h^2)$,容易求出它的增长因子

$$G=[1-2\beta(1-\cos\omega h)]-\mathrm{i}\alpha\sin\omega h,$$

其中

$$\alpha=a\frac{\tau}{h},\quad \beta=b\frac{\tau}{h^2},$$

由此容易得到差分格式(12.3.18)的稳定性条件是

$$\beta\leqslant\frac{1}{2}\quad 和\quad \alpha^2\leqslant 2\beta.$$

容易推导出,用中心显式格式求解微分方程

$$\frac{\partial u}{\partial t}+a\frac{\partial u}{\partial x}=\left(b-\frac{\tau}{2}a^2\right)\frac{\partial^2 u}{\partial x^2},\tag{12.3.19}$$

当$\tau\to 0$时,方程(12.3.19)与对流扩散方程(12.3.6)是相同的.但是,在计算中$\tau\neq 0$,因此用差分格式(12.3.18)来计算方程(12.3.6)时,导致扩散效应的减少.特别当$\frac{\tau}{2}a^2$接近于b时更为显著.为使扩散效应不致于减少,由此导出**修正显式中心格式**,即

$$\frac{u_j^{n+1}-u_j^n}{\tau}+\frac{a}{2h}(u_{j+1}^n-u_{j-1}^n)=\left(b+\frac{\tau}{2}a^2\right)\frac{1}{h^2}(u_{j+1}^n-2u_j^n+u_{j-1}^n),$$

(12.3.20)

此格式的截断误差为$O(\tau+h^2)$,稳定性条件为

$$\alpha^2+2\beta\leqslant 1.$$

在实际应用中,经常碰到$|a|\gg b$的情况,即所谓对流占优的情况,这种情况中心差分格式将不能很好地进行计算,而迎风格式则大致可以应用.为确定起见,令$a>0$,则逼近对流方程(12.3.6)的迎风显式格式为

$$\frac{u_j^{n+1}-u_j^n}{\tau}+a\frac{u_j^n-u_{j-1}^n}{h}=b\frac{u_{j+1}^n-2u_j^n+u_{j-1}^n}{h^2},$$

(12.3.21)

其截断误差为 $O(\tau+h)$，增长因子

$$G=[1-(2\beta+\alpha)(1-\cos\omega h)]-\alpha i\sin\omega h.$$

由此可以推出差分格式(12.3.21)的稳定条件为

$$\alpha+2\beta\leqslant 1.$$

迎风格式的截断误差均为一阶，为提高其精度，同样可以计算对流占优的扩散问题还有指数格式及萨马尔斯基(Samarskii)格式.

显式差分格式稳定性是有条件的，因此相应的隐式格式很受重视，特别是克拉克-尼科尔森格式，相应的指数格式被称为有限解析格式，也很受重视.

常用的差分格式见表 12.5.

表 12.5

名　称	差分格式与截断误差	稳定条件
中心显式格式	$\frac{1}{\tau}(u_j^{n+1}-u_j^n)+\frac{a}{2h}(u_{j+1}^n-u_{j-1}^n)$ $=b\frac{1}{h^2}(u_{j+1}^n-2u_j^n+u_{j-1}^n)$ $E=O(\tau+h^2)$	$\beta\leqslant\frac{1}{2}$ $\alpha^2\leqslant 2\beta$
修正中心格式	$\frac{1}{\tau}(u_j^{n+1}-u_j^n)+\frac{a}{2h}(u_{j+1}^n-u_{j-1}^n)$ $=\left(b+\frac{a^2}{2}\tau\right)\frac{1}{h^2}(u_{j+1}^n-2u_j^n+u_{j-1}^n)$ $E=O(\tau+h^2)$	$\alpha^2+2\beta\leqslant 1$
杜福特-弗兰克尔显式格式	$\frac{1}{2\tau}(u_j^{n+1}-u_j^{n-1})+\frac{a}{2h}(u_{j+1}^n-u_{j-1}^n)$ $=b\frac{1}{h^2}[u_{j+1}^n-(u_j^{n+1}+u_j^{n-1})+u_{j-1}^n]$ $E=O(\tau^2/h^2)+O(\tau^2+h^2)$	$\alpha\leqslant 1$
迎风显式格式	$\frac{1}{\tau}(u_j^{n+1}-u_j^n)+\frac{a}{h}(u_j^n-u_{j-1}^n)$ $=b\frac{1}{h^2}(u_{j+1}^n-2u_j^n+u_{j-1}^n)$ $E=O(\tau+h)$	$\alpha+2\beta\leqslant 1$

续表

名　称	差分格式与截断误差	稳定条件
指数型差分格式	$\frac{u_j^{n+1}-u_j^n}{\tau}+a\frac{u_{j+1}^n-u_{j-1}^n}{2h}=b\sigma\frac{u_{j+1}^n-2u_j^n+u_{j-1}^n}{h^2}$ $\sigma=\frac{ah}{2b}\coth\left(\frac{ah}{2b}\right)$ $E=O(\tau+h^2)$	$b\sigma\frac{\tau}{h^2}\leqslant\frac{1}{2}$
萨马尔斯基格式	$\frac{u_j^{n+1}-u_j^n}{\tau}+a\frac{u_j^n-u_{j-1}^n}{h}=\frac{b}{1+R}\frac{u_{j+1}^n-2u_j^n+u_{j-1}^n}{h^2}$ $R=\frac{ah}{2b}$ $E=O(\tau+h^2)$	$\left(\frac{ah}{2}+\frac{b}{1+\frac{ah}{2b}}\right)\frac{\tau}{h^2}\leqslant\frac{1}{2}$
全隐格式	$\frac{1}{\tau}(u_j^{n+1}-u_j^n)+\frac{a}{2h}(u_{j+1}^{n+1}-u_{j-1}^{n+1})$ $=b\frac{1}{h^2}(u_{j+1}^{n+1}-2u_j^{n+1}+u_{j-1}^{n+1})$ $E=O(\tau+h^2)$	绝对稳定
克拉克-尼科尔森格式	$\frac{1}{\tau}(u_j^{n+1}-u_j^n)+\frac{a}{2h}[(u_{j+1}^{n+1}-u_{j-1}^{n+1})+$ $(u_{j+1}^n-u_{j-1}^n)]$ $=\frac{b}{2h^2}\Big[(u_{j+1}^{n+1}-2u_j^{n+1}+u_{j-1}^{n+1})+$ $(u_{j+1}^n-2u_j^n+u_{j-1}^n)\Big]$ $E=O(\tau^2+h^2)$	绝对稳定
隐式迎风格式	$\frac{1}{\tau}(u_j^{n+1}-u_j^n)+\frac{a}{h}(u_j^{n+1}-u_{j-1}^{n+1})$ $=b\frac{1}{h^2}(u_{j+1}^{n+1}-2u_j^{n+1}+u_{j-1}^{n+1})$ $E=O(\tau+h)$	绝对稳定

续表

名　称	差分格式与截断误差	稳定条件
隐式抬数格式	$\frac{u_j^{n+1}-u_j^n}{\tau}+a\frac{u_{j+1}^n-u_{j-1}^n}{2h}$ $=b\sigma\frac{u_{j+1}^{n+1}-2u_j^{n+1}+u_{j-1}^{n+1}}{h^2}$ $E=O(\tau+h^2)$	绝对稳定
	$\frac{u_j^{n+1}-u_j^n}{\tau}+a\frac{u_j^{n+1}-u_{j-1}^{n+1}}{h}$ $=\frac{b}{1+R}\frac{u_{j+1}^{n+1}-2u_j^{n+1}+u_{j-1}^{n+1}}{h^2}$ $E=O(\tau+h^2)$	绝对稳定

12.3.4　二维扩散方程的差分方法

考虑二维扩散方程的初边值混合问题(12.3.8)的差分格式. 首先对区域

$$\Omega=\{(x,y,t)\mid 0\leqslant x,y\leqslant l,t\geqslant 0\}$$

进行网格剖分. 为简单起见, x 方向和 y 方向的空间步长皆取 h, 时间步长为 τ, 这样网格节点可以记为 $x_j=jh$, $y_k=kh$, $t_n=n\tau$, 其中 $j,k=0,1,\cdots,M$, $Mh=l$, $n\geqslant 0$. 引进记号

$$\delta_x^2u_{jk}^n=u_{j+1,k}^n-2u_{jk}^n+u_{j-1,k}^n,$$

$$\delta_y^2u_{jk}^n=u_{j,k+1}^n-2u_{jk}^n+u_{j,k-1}^n,$$

其中 u_{ij} 是取在网格点上的函数.

利用差商替代微商的办法, 可以直接写出问题(12.3.8)中扩散方程的一些差分格式. 一维扩散方程显式格式的直接推广是

$$\frac{u_{jk}^{n+1}-u_{jk}^n}{\tau}=\frac{b}{h^2}(\delta_x^2u_{jk}^n+\delta_y^2u_{jk}^n),\qquad(12.3.22)$$

容易求得此差分格式的截断误差同一维情况一样, 即为 $O(\tau+$

h^2). 仿一维的方法,可以算出式(12.3.22)的增长因子

$$G = 1 - 4b\frac{\tau}{h^2}\sin^2\frac{\omega_1 h}{2} - 4b\frac{\tau}{h^2}\sin^2\frac{\omega_2 h}{2},$$

所以差分格式(12.3.22)的稳定性条件是

$$b\frac{\tau}{h^2} \leqslant \frac{1}{4}.$$

在一维情形,稳定性要求 $b\frac{\tau}{h^2}\leqslant\frac{1}{2}$,由例 12.3.1 可以看出,显式格式的时间步长 τ 是克拉克-尼科尔森格式的时间步长的 1/20 时才能计算出相近的数值结果,而二维扩散方程的显式格式的步长要求更严,所以在实际计算中采用显式格式是不可取的.

一维全隐格式和克拉克-尼科尔森格式的直接推广为

$$\frac{u_{jk}^{n+1} - u_{jk}^{n}}{\tau} = \frac{b}{h^2}(\delta_x^2 u_{jk}^{n+1} + \delta_y^2 u_{jk}^{n+1}), \qquad (12.3.23)$$

和

$$\frac{u_{jk}^{n+1} - u_{jk}^{n}}{\tau} = \frac{b}{2h^2}(\delta_x^2 u_{jk}^{n+1} + \delta_y^2 u_{jk}^{n+1}) + \frac{b}{2h^2}(\delta_x^2 u_{jk}^{n} + \delta_y^2 u_{jk}^{n}). \qquad (12.3.24)$$

这两个格式的截断误差分别为 $O(\tau+h^2)$ 和 $O(\tau^2+h^2)$,并都是绝对稳定的. 但是,用差分格式(12.3.23)和(12.3.24)来求解问题(12.3.8),其形成的线性代数方程组的系数矩阵已不是三对角线矩阵,所以不能用追赶法求解. 由于在每一时间层上,求解线性代数方程组是费事的,所以隐式格式在实际计算中也是不方便的.

经常使用的格式,一般都具有绝对稳定、有合理的精度、得到的代数方程组容易求解等优点,交替方向隐式格式就是其中之一. 如,逼近问题(12.3.8)中扩散方程的**皮斯曼-瑞奇福尔德(Peaceman-Rachford)格式**



$$\begin{cases} \dfrac{u_{jk}^{n+\frac{1}{2}} - u_{jk}^{n}}{\tau/2} = \dfrac{b}{h^2}\delta_x^2 u_{jk}^{n+\frac{1}{2}} + \dfrac{b}{h^2}\delta_y^2 u_{jk}^{n}, \\ \dfrac{u_{jk}^{n+1} - u_{jk}^{n+\frac{1}{2}}}{\tau/2} = \dfrac{b}{h^2}\delta_x^2 u_{jk}^{n+\frac{1}{2}} + \dfrac{b}{h^2}\delta_y^2 u_{jk}^{n+1}, \end{cases} \tag{12.3.25}$$

它的截断误差为 $O(\tau^2+h^2+h^2)$，由式(12.3.25)可知，在一个时间步长上，只要在 x 方向和 y 方向用一系列追赶法就可以解出方程组，因而计算容易. 类似地，有**近似分裂方法**、**局部一维方法**都有同样解法，具体格式见表 12.6.

表 12.6

名 称	差分格式与截断误差	稳定条件
皮斯曼-瑞奇福尔德交替方向隐式格式	$\dfrac{u_{ij}^{n+\frac{1}{2}}-u_{ij}^{n}}{\tau/2}=\dfrac{b}{h^2}(\delta_x^2 u_{ij}^{n+\frac{1}{2}}+\delta_y^2 u_{ij}^{n})$ $\dfrac{u_{ij}^{n+1}-u_{ij}^{n+\frac{1}{2}}}{\tau/2}=\dfrac{b}{h^2}(\delta_x^2 u_{ij}^{n+\frac{1}{2}}+\delta_y^2 u_{ij}^{n+1})$ $E=O(\tau^2+h^2+h^2)$	绝对稳定
道格拉斯-瑞奇福尔德格式	$\dfrac{u_{ij}^{n+\frac{1}{2}}-u_{ij}^{n}}{\tau}=\dfrac{b}{h^2}[\delta_x^2 u_{ij}^{n+\frac{1}{2}}+\delta_y^2 u_{ij}^{n}]$ $\dfrac{u_{ij}^{n+1}-u_{ij}^{n+\frac{1}{2}}}{\tau}=\dfrac{b}{h^2}[\delta_y^2 u_{ij}^{n+1}-\delta_y^2 u_{ij}^{n}]$ $E=O(\tau+h^2+h^2)$	绝对稳定
道格拉斯格式	$\left(1-\dfrac{b}{2}\lambda\delta_x^2\right)\dfrac{u_{jk}^{n+\frac{1}{2}}-u_{jk}^{n}}{\tau}=\dfrac{b}{h^2}(\delta_x^2+\delta_y^2)u_{jk}^{n}$ $\dfrac{u_{jk}^{n+1}-u_{jk}^{n+\frac{1}{2}}}{\tau}=b\dfrac{1}{h^2}\delta_y^2(u_{jk}^{n+1}-u_{jk}^{n})$ $E=O(\tau^2+h^2+h^2)$	绝对稳定
D'yakonov 格式	$\left(1-\dfrac{b}{2}\lambda\delta_x^2\right)u_{jk}^{n+\frac{1}{2}}=\left(1+\dfrac{b}{2}\lambda\delta_x^2\right)\times\left(1+\dfrac{b}{2}\lambda\delta_y^2\right)u_{jk}^{n}$ $\left(1-\dfrac{b}{2}\lambda\delta_y^2\right)u_{jk}^{n+1}=u_{jk}^{n+\frac{1}{2}}$ $E=O(\tau^2+h^2+h^2)$	绝对稳定

续表

名　称	差分格式与截断误差	稳定条件
局部一维格式	$\frac{u_{jk}^{n+\frac{1}{2}}-u_{jk}^n}{\tau}=\frac{b}{h^2}\delta_x^2\left(\frac{u_{jk}^{n+\frac{1}{2}}+u_{jk}^n}{2}\right)$ $\frac{u_{jk}^{n+1}-u_{jk}^{n+\frac{1}{2}}}{\tau}=\frac{b}{h^2}\delta_y^2\left(\frac{u_{jk}^{n+1}+u_{jk}^{n+\frac{1}{2}}}{2}\right)$ $E=O(\tau^2+h^2+h^2)$	绝对稳定
预估-校正格式	$\frac{u_{jk}^{n+\frac{1}{4}}-u_{jk}^n}{\tau/2}=\frac{b}{h^2}\delta_x^2u_{jk}^{n+\frac{1}{4}}$ $\frac{u_{jk}^{n+\frac{1}{2}}-u_{jk}^{n+\frac{1}{4}}}{\tau/2}=\frac{b}{h^2}\delta_y^2u_{jk}^{n+\frac{1}{2}}$ $\frac{u_{jk}^{n+1}-u_{jk}^n}{\tau}=\frac{1}{h^2}(\delta_x^2+\delta_y^2)u_{jk}^{n+\frac{1}{2}}$ $E=O(\tau^2+h^2+h^2)$	

12.4 有限元方法

解椭圆型边值问题的有限元方法,可以看成是变分原理与函数分片多项式插值逼近方法的结合.在几何条件和物理条件比较复杂的问题中,有限元方法比有限差分方法有更广泛的适应性.

12.4.1 椭圆型边值问题的变分原理

考虑 Oxy 平面的区域 Ω 上如下的边值问题:

$$\begin{cases} -\left[\frac{\partial}{\partial x}\left(k\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(k\frac{\partial u}{\partial y}\right)\right]=f, & (x,y)\in\Omega, \quad (12.4.1)\\ u=0, & (x,y)\in\partial\Omega_1, \quad (12.4.2)\\ k\frac{\partial u}{\partial n}+\sigma u=g, & (x,y)\in\partial\Omega_2, \quad (12.4.3)\end{cases}$$

其中 Ω 的边界$\partial\Omega$ 分为互不重叠的两部分$\partial\Omega_1$ 和$\partial\Omega_2$,n 是$\partial\Omega$ 上的外法线方向,f,g 和 k,σ 都是给定的函数,k 是一阶偏导数连续的

函数，且 $k(x,y)\geqslant k_0>0$，σ 是连续函数，且 $\sigma(x,y)\geqslant 0$.

对应于边值问题(12.4.1)～(12.4.3)，有下面的变分问题，其中的函数集合

$$V=\{v\mid v\in C^1(\Omega),v|_{\partial\Omega_1}=0\}. \tag{12.4.4}$$

泛函 F 和双线性泛函 D 满足

$$F(v)=\iint_{\Omega}fv\mathrm{d}x\mathrm{d}y+\int_{\partial\Omega_2}gv\mathrm{d}s, \tag{12.4.5}$$

$$D(u,v)=\iint_{\Omega}k\left(\frac{\partial u}{\partial x}\frac{\partial v}{\partial x}+\frac{\partial u}{\partial y}\frac{\partial v}{\partial y}\right)\mathrm{d}x\mathrm{d}y+\int_{\partial\Omega_2}\sigma uv\mathrm{d}s. \tag{12.4.6}$$

伽辽金变分问题：

求 $u\in V$，使得

$$D(u,v)-F(v)=0 \tag{12.4.7}$$

对一切 $v\in V$ 成立.

里茨变分问题：

求 $u\in V$，使得

$$J(u)\leqslant J(v) \tag{12.4.8}$$

对一切 $v\in V$ 成立. 其中

$$J(v)=\frac{1}{2}D(v,v)-F(v). \tag{12.4.9}$$

如果把边值问题(12.4.1)～(12.4.3)的物理意义理解为弹性薄膜的平衡问题，那么伽辽金变分问题(12.4.7)和里茨变分问题(12.4.8)可以分别理解为对应平衡问题的**虚功原理**和**最小势能原理**. 它们以不同的形式反映同一物理现象.

定理 12.4.1 如果 $u\in C^2(\Omega)$是边值问题(12.4.1)～(12.4.3)的解，则 u 是伽辽金变分问题(12.4.7)的解. 反之，如果 u 是伽辽金变分问题(12.4.7)的解，且 $u\in C^2(\Omega)$，则 u 是边值问题(12.4.1)～(12.4.3)的解. 如果 u 是伽辽金变分问题(12.4.7)的解，则 u 是里茨变分问题(12.4.8)的解，反之亦然.

事实上，在变分问题中，式(12.4.4)所表示的函数空间 V 中，$v\in C^1(\Omega)$的条件可以改为 v 及 v 的一阶偏导数都是平方可积的，这样，变分问题就存在惟一的解，而且定理 12.4.1 成立. 这涉及用索伯列夫空间(Sobolev space)理论讨论微分方程及有限元方法的理论问题.

如果在方程(12.4.1)中的系数 $k(x,y)$有间断，它的间断线将 Ω 分为若干个子域，为了简化讨论，设间断线 Γ 将区域Ω 分为子域Ω_1 和 Ω_2，并规定了 Γ 的法线方向 $\boldsymbol{n}$，如图 12.17 所示. 记 Γ 上的函数值 $(\cdot)^i$ 为该函数当自变量从 Ω_i 趋于 Γ 时的极限值($i=1,2$).

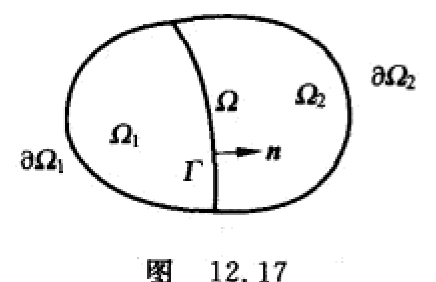


图 12.17

间断系数边值问题的一个例子是：

$$
\begin{cases}
-\left[\dfrac{\partial}{\partial x}\left(k\dfrac{\partial u}{\partial x}\right)+\dfrac{\partial}{\partial y}\left(k\dfrac{\partial u}{\partial y}\right)\right]=f, \quad (x,y)\in\Omega, \\
u=0, \quad (x,y)\in\partial\Omega_1, \\
k\dfrac{\partial u}{\partial n}+\sigma u=g, \quad (x,y)\in\partial\Omega_2, \\
u^1=u^2, \quad (x,y)\in\Gamma, \qquad (12.4.10) \\
\left(k\dfrac{\partial u}{\partial n}\right)^1=\left(k\dfrac{\partial u}{\partial n}\right)^2, \quad (x,y)\in\Gamma. \qquad (12.4.11)
\end{cases}
$$

其中式(12.4.10)和式(12.4.11)保证 u 和 $k\dfrac{\partial u}{\partial n}$在 Γ 上的连续性.

定理 12.4.2 在系数 k 间断的情况下，边值问题对应的变分问题仍是式(12.4.7)和式(12.4.8).

根据变分原理用有限元方法处理间断系数的边值问题，与非间断系数的边值问题的处理方法是相同的.

12.4.2 三角形线性元

用有限元方法解二维椭圆型边值问题，最常用和最简单的方法是对区域 Ω 做三角形剖分，在每个三角形单元上做线性插值.

将 Ω 剖分为一组三角形的组合，Ω 的边界就以折线替代，如图 12.18 所示，设这组三角形最大边长为 h.

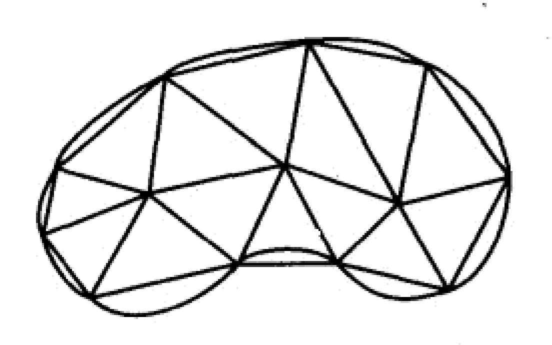


图 12.18

每个剖分后的三角形称为一个**单元**，它的顶点称为**节点**. 对单元和节点进行编号. 设有 NE 个单元 $e_k(k=1,2,\cdots,NE)$；有 NP 个节点 P_i，其坐标为$(x_i,y_i)(i=1,2,\cdots,NP)$. 剖分时注意不要出现大钝角的三角形，同时可以根据物理量的变化布置节点的疏密，在关键部位可加密，其他部位可放疏，但疏密的过渡要避免变化太快.

用区域 Ω 上的分片线性插值函数近似 u，这样的函数在 Ω 上是连续的，而且在每个单元 e_k 上都是 x,y 的一次多项式，把这样的分片线性多项式记为 $u_h(x,y)$. 为了在每个单元 e_k 上列出 $u_h(x,y)$的表达式，设 e 是任意一个单元，其顶点为 P_i,P_j,P_m，并规定 i,j,m 按逆时针顺序排列，如图 12.19.

设 $u_h(x,y)$在三角形单元 e 的三个顶点上的值为

$$u_i=u_h(x_i,y_i),\quad u_j=u_h(x_j,y_j),\quad u_m=u_h(x_m,y_m).$$

三角形单元 e 的面积为

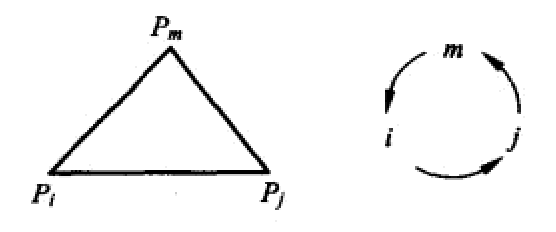


图 12.19

$$\Delta_e = \frac{1}{2}\begin{vmatrix} x_i & y_i & 1 \\ x_j & y_j & 1 \\ x_m & y_m & 1 \end{vmatrix}.$$

这样在单元 e 上，有

$$u_h(x,y) = N_i(x,y)u_i + N_j(x,y)u_j + N_m(x,y)u_m. \tag{12.4.12}$$

其中 $N_i(x,y)$，$N_j(x,y)$和 $N_m(x,y)$称为**线性插值基函数**(basis function of linear interpolation)，函数式为

$$\begin{cases} N_i(x,y) = \dfrac{1}{2\Delta_e}(a_i x + b_i y + c_i), \\ N_j(x,y) = \dfrac{1}{2\Delta_e}(a_j x + b_j y + c_j), \\ N_m(x,y) = \dfrac{1}{2\Delta_e}(a_m x + b_m y + c_m), \end{cases} \tag{12.4.13}$$

其中

$$a_i = y_j - y_m, \quad a_j = y_m - y_i, \quad a_m = y_i - y_j;$$

$$b_i = x_m - x_j, \quad b_j = x_i - x_m, \quad b_m = x_j - x_i;$$

$$c_i = \begin{vmatrix} x_j & y_j \\ x_m & y_m \end{vmatrix}, \quad c_j = \begin{vmatrix} x_m & y_m \\ x_i & y_i \end{vmatrix}, \quad c_m = \begin{vmatrix} x_i & y_i \\ x_j & y_j \end{vmatrix}.$$

线性插值基函数有如下的性质：

$$N_k(x_l, y_l) = \begin{cases} 0, & k \neq l \\ 1, & k = l \end{cases}, \quad (k,l = i,j,m), \tag{12.4.14}$$

$$\begin{cases} N_i + N_j + N_m = 1, \\ x_i N_i + x_j N_j + x_m N_m = x, \\ y_i N_i + y_j N_j + y_m N_m = y. \end{cases} \tag{12.4.15}$$

作为例子，考虑边值问题(12.4.1)～(12.4.3)中系数 $k(x,y)\equiv 1$ 的情形. 从变分问题(12.4.7)出发，以分片线性插值函数近似 u，可得到**近似变分问题**(approximate variational problem)：

求 $u_h \in V_h$，使得

$$D(u_h, v_h) - F(v_h) = 0, \tag{12.4.16}$$

对一切 $v_h \in V_h$ 成立，其中 $D(\cdot,\cdot)$ 与 $F(\cdot)$ 仍如式(12.4.6)和式(12.4.5)所示，V_h 则是所有在 $\partial\Omega_1$ 上为零的分片线性函数的集合.

为了将近似变分问题(12.4.16)逐个单元计算，记第 n 个单元为 e_n，其边界为 ∂e_n. 若 ∂e_n 中至少有一条边在 Ω 的边界 $\partial\Omega_2$ 上，则把这段边界记为 γ_n. 若 ∂e_n 的三条边都不在 $\partial\Omega_2$ 上，规定 γ_n 为空集，在其上积分值为零. 这样，近似变分问题(12.4.16)可写成：

求 $u_h \in V_h$，使得

$$\sum_{n=1}^{NE}\left[\iint\limits_{e_n}\left(\frac{\partial u_h}{\partial x}\frac{\partial v_h}{\partial x}+\frac{\partial u_h}{\partial y}\frac{\partial v_h}{\partial y}\right)\mathrm{d}x\mathrm{d}y+\int_{\gamma_n}\sigma u_h v_h \mathrm{d}s\right]$$

$$=\sum_{n=1}^{NE}\left[\iint\limits_{e_n} f v_h \mathrm{d}x\mathrm{d}y+\int_{\gamma_n} g v_h \mathrm{d}s\right] \tag{12.4.17}$$

对一切 $v_h \in V_h$ 成立.

里茨变分问题(12.4.8)的近似变分问题可以类似地写出. 在边值问题中若 $k(x,y)$ 不是常数，以上问题的提法只需稍作修改.

为了逐元计算(12.4.17)中的积分，设 e_n 为 $\Delta P_i P_j P_m$，其顶点上 u_h, v_h 之值分别为 u_i, u_j, u_m 和 v_i, v_j, v_m，并记向量

$$\boldsymbol{u}_{e_n} = (u_i,\ u_j,\ u_m)^{\mathrm{T}},$$

$$\boldsymbol{v}_{e_n} = (v_i,\ v_j,\ v_m)^{\mathrm{T}},$$

$$\boldsymbol{N} = (N_i,\ N_j,\ N_m).$$

则在 e_n 上有

$$u_h = \boldsymbol{N}\boldsymbol{u}_{e_n}, \quad v_h = \boldsymbol{N}\boldsymbol{v}_{e_n},$$

$$\begin{pmatrix} \dfrac{\partial u_h}{\partial x} \\ \dfrac{\partial u_h}{\partial y} \end{pmatrix} = \boldsymbol{B}\boldsymbol{u}_{e_n}, \quad \begin{pmatrix} \dfrac{\partial v_h}{\partial x} \\ \dfrac{\partial v_h}{\partial y} \end{pmatrix} = \boldsymbol{B}\boldsymbol{v}_{e_n},$$

$$\boldsymbol{B} = \frac{1}{2\Delta_e}\begin{pmatrix} a_i & a_j & a_m \\ b_i & b_j & b_m \end{pmatrix}.$$

为了方便起见，如果 γ_n 不是空集，设 γ_n 是$\overline{P_iP_j}$边（其他情形以下计算类似），引入参数 t 为$\overline{P_iP_j}$上的弧长，对应点 P_i 有 $t=0$，对应点 P_j 有 $t=l$，这样可得

$$N_i\Big|_{\overline{P_iP_j}} = 1-\frac{t}{l}, \quad N_j\Big|_{\overline{P_iP_j}} = \frac{t}{l}, \quad N_m\Big|_{\overline{P_iP_j}} = 0.$$

将这些式子代入等式(12.4.17)左端的积分，可得

$$\iint\limits_{e_n}\left(\frac{\partial u_h}{\partial x}\frac{\partial v_h}{\partial x}+\frac{\partial u_h}{\partial y}\frac{\partial v_h}{\partial y}\right)\mathrm{d}x\mathrm{d}y+\int_{\gamma_n}\sigma u_h v_h\,\mathrm{d}x = \boldsymbol{v}_{e_n}^{\mathrm{T}}\boldsymbol{K}_{e_n}\boldsymbol{u}_{e_n},$$

其中 $\boldsymbol{K}_{e_n}$ 为**单元刚度矩阵**(element stiffness matrix)：

$$\boldsymbol{K}_{e_n} = \bar{\boldsymbol{K}}_{e_n} + \tilde{\boldsymbol{K}}_{e_n},$$

$$\bar{\boldsymbol{K}}_{e_n} = \begin{pmatrix} \bar{k}_{ii} & \bar{k}_{ij} & \bar{k}_{im} \\ \bar{k}_{ji} & \bar{k}_{jj} & \bar{k}_{jm} \\ \bar{k}_{mi} & \bar{k}_{mj} & \bar{k}_{mm} \end{pmatrix},$$

$$\bar{k}_{st} = \frac{1}{4\Delta_{e_n}}(a_s a_t + b_s b_t) \quad (s,t=i,j,m).$$

当 γ_n 为空集时，$\tilde{K}_{e_n}=\mathbf{0}$. 否则

$$\tilde{\boldsymbol{K}}_{e_n} = \begin{pmatrix} \tilde{k}_{ii} & \tilde{k}_{ij} & \tilde{k}_{im} \\ \tilde{k}_{ji} & \tilde{k}_{jj} & \tilde{k}_{jm} \\ \tilde{k}_{mi} & \tilde{k}_{mj} & \tilde{k}_{mm} \end{pmatrix},$$

$$\tilde{k}_{ii} = \int_0^l \sigma\left(1 - \frac{t}{l}\right)^2 \mathrm{d}t,$$

$$\tilde{k}_{ij} = \tilde{k}_{ji} = \int_0^l \sigma\left(1 - \frac{t}{l}\right)\frac{t}{l}\mathrm{d}t,$$

$$\tilde{k}_{jj} = \int_0^l \sigma\left(\frac{t}{l}\right)^2 \mathrm{d}t,$$

$$\tilde{k}_{mi} = \tilde{k}_{im} = \tilde{k}_{mj} = \tilde{k}_{jm} = \tilde{k}_{mm} = 0.$$

等式(12.4.17)右端的积分可得

$$\iint_{e_n} f v_h \mathrm{d}x\mathrm{d}y + \int_{\gamma_n} g v_h \mathrm{d}s = \boldsymbol{v}_{e_n}^{\mathrm{T}} \boldsymbol{F}_{e_n},$$

其中 $\boldsymbol{F}_{e_n}$ 为单元荷载向量(element load vector):

$$\boldsymbol{F}_{e_n} = \overline{\boldsymbol{F}}_{e_n} + \widetilde{\boldsymbol{F}}_{e_n},$$

$$\overline{\boldsymbol{F}}_{e_n} = (\overline{\boldsymbol{F}}_i, \quad \overline{\boldsymbol{F}}_j, \quad \overline{\boldsymbol{F}}_m)^{\mathrm{T}},$$

$$\overline{\boldsymbol{F}}_s = \iint_{e_n} N_s f \mathrm{d}x\mathrm{d}y \quad (s = i, j, m).$$

当 γ_n 为空集时,$\widetilde{\boldsymbol{F}}_{e_n} = \boldsymbol{0}$,否则

$$\widetilde{\boldsymbol{F}}_{e_n} = \left[\int_0^l \left(1 - \frac{t}{l}\right) g \mathrm{d}t, \quad \int_0^l \frac{t}{l} g \mathrm{d}t, \quad 0\right]^{\mathrm{T}}.$$

为了把式(12.4.17)中各单元的积分叠加起来.令向量

$$\boldsymbol{u} = (u_1, u_2, \cdots, u_{NP})^{\mathrm{T}},$$

$$\boldsymbol{v} = (v_1, v_2, \cdots, v_{NP})^{\mathrm{T}}.$$

设单元 $e_n = \Delta P_i P_j P_m$,则

$$\boldsymbol{v}_{e_n} = (v_i, v_j, v_m)^{\mathrm{T}} = \boldsymbol{C}_{e_n} \boldsymbol{v},$$

其中 $\boldsymbol{C}_{e_n}$ 是一个 $3 \times NP$ 的矩阵.设 P_i 点在节点总编号中序号为 k_i,则 $\boldsymbol{C}_{e_n}$ 的第 1 行第 k_i 个元素为 1,其余元素为 0.同理 $\boldsymbol{C}_{e_n}$ 的第 2,3 行只有一个非零元素,其值为 1,它的位置由 P_j, P_m 在节点总编号中的序号决定.同理.

$$\boldsymbol{u}_{e_n} = (u_i, u_j, u_m)^{\mathrm{T}} = \boldsymbol{C}_{e_n} \boldsymbol{u}.$$

式(12.4.17)左端积分的叠加得到

$$\sum_{n=1}^{NE} \boldsymbol{v}_{e_n}^{\mathrm{T}} \boldsymbol{K}_{e_n} \boldsymbol{u}_{e_n} = \boldsymbol{v}^{\mathrm{T}} \boldsymbol{K} \boldsymbol{u},$$

其中 $\boldsymbol{K}$ 称为**刚度矩阵**(stiffness matrix),也称为**总刚度矩阵**. 它是 $NP \times NP$ 的矩阵,表达式为

$$\boldsymbol{K} = \sum_{n=1}^{NE} \boldsymbol{C}_{e_n}^{\mathrm{T}} \boldsymbol{K}_{e_n} \boldsymbol{C}_{e_n}.$$

式(12.4.17)右端积分的叠加得到

$$\sum_{n=1}^{NE} \boldsymbol{v}_{e_n}^{\mathrm{T}} \boldsymbol{F}_{e_n} = \boldsymbol{v}^{\mathrm{T}} \boldsymbol{F},$$

其中 $\boldsymbol{F}$ 称为**荷载向量**(load vector),也称**总荷载向量**. 它是 NP 维的向量,表达式为

$$\boldsymbol{F} = \sum_{n=1}^{NE} \boldsymbol{C}_{e_n}^{\mathrm{T}} \boldsymbol{F}_{e_n}.$$

式(12.4.17)最后成为等式

$$\boldsymbol{v}^{\mathrm{T}}(\boldsymbol{K}\boldsymbol{u} - \boldsymbol{F}) = \boldsymbol{0}. \tag{12.4.18}$$

在实际计算总刚度矩阵 $\boldsymbol{K}$ 时,应注意它由 $\boldsymbol{C}_{e_n}^{\mathrm{T}} \boldsymbol{K}_{e_n} \boldsymbol{C}_{e_n}$ ($n=1,2,\cdots,NE$)叠加而得. $\boldsymbol{C}_{e_n}^{\mathrm{T}} \boldsymbol{K}_{e_n} \boldsymbol{C}_{e_n}$ 是 $NP \times NP$ 的矩阵,但它只有 9 个非零元素,这 9 个元素就是 $\boldsymbol{K}_{e_n}$ 的 9 个元素,它们在 $NP \times NP$ 矩阵中的位置由 P_i, P_j, P_m 在节点总编号中的序号所确定,所以 $\boldsymbol{C}_{e_n}^{\mathrm{T}} \boldsymbol{K}_{e_n} \boldsymbol{C}_{e_n}$ 只是由 3×3 矩阵 $\boldsymbol{K}_{e_n}$ "扩大"为 $NP \times NP$ 的矩阵. 例如,若计算 e_n 时,P_i, P_j 和 P_m 在节点总编号的序号分别是 101,102,95,则有

$$\boldsymbol{C}_{e_n}^{\mathrm{T}} \boldsymbol{K}_{e_n} \boldsymbol{C}_{e_n} = \begin{pmatrix} & \vdots & & \vdots & \vdots & \\ \cdots & k_{mm} & \cdots & k_{mi} & k_{mj} & \cdots \\ & \vdots & & \vdots & \vdots & \\ \cdots & k_{im} & \cdots & k_{ii} & k_{ij} & \cdots \\ \cdots & k_{jm} & \cdots & k_{ji} & k_{jj} & \cdots \\ & \vdots & & \vdots & \vdots & \end{pmatrix} \begin{matrix} \\ \text{第 95 行} \\ \\ \text{第 101 行} \\ \text{第 102 行} \\ \\ \end{matrix}$$

$$\qquad\qquad \text{第 95 列} \quad \text{第 101 列} \quad \text{第 102 列}$$

同理，$C_{e_n}^{T}F_{e_n}$ 是 NP 维向量，它只有三个非零元素，所以它是三维向量 F_{e_n} 的"扩大". 在上面的例子中

$$C_{e_n}^{T}F_{e_n}=\begin{pmatrix}\vdots\\ F_m\\ \vdots\\ F_i\\ F_j\\ \vdots\end{pmatrix}\begin{matrix}\\ 95\\ \\ 101\\ 102\\ \\ \end{matrix},$$

为了叙述方便，假设节点编号时，把边界 $\partial\Omega_1$（给出本质边界条件的部分）的节点排在最前面，即 $P_1,P_2,\cdots,P_l$，其他节点为 $P_{l+1},P_{l+2},\cdots,P_{NP}$. 在近似变分问题(12.4.16)中，$u_h,v_h$ 都属于 V_h，即 v_h 及 u_h 在节点 $P_1,P_2,\cdots,P_l$ 上的值都为 0. 这样 NP 维向量 $\boldsymbol{u}$ 和 $\boldsymbol{v}$ 的前 l 个分量都为 0，即

$$\boldsymbol{v}=(0,0,\cdots,0,v_{l+1},v_{l+2},\cdots,v_{NP})^{\mathrm{T}},$$

$$\boldsymbol{u}=(0,0,\cdots,0,u_{l+1},u_{l+2},\cdots,u_{NP})^{\mathrm{T}}.$$

这样，近似变分问题(12.4.16)或(12.4.17)化为如下的**离散问题**(discrete problem)：

求 $\boldsymbol{u}=(0,0,\cdots,0,u_{l+1},u_{l+2},\cdots,u_{NP})^{\mathrm{T}}$，

使得
$$\boldsymbol{v}^{\mathrm{T}}(\boldsymbol{K}\boldsymbol{u}-\boldsymbol{F})=\boldsymbol{0} \tag{12.4.19}$$

对任意的 $\boldsymbol{v}=(0,0,\cdots,0,v_{l+1},v_{l+2},\cdots,v_{NP})^{\mathrm{T}}$ 成立.

以上离散问题可以化为求解线性代数方程组的问题. 用分块矩阵记号记

$$\boldsymbol{K}=\begin{pmatrix}\boldsymbol{K}_{11} & \boldsymbol{K}_{12}\\ \boldsymbol{K}_{21} & \boldsymbol{K}_{22}\end{pmatrix},$$

$$\boldsymbol{v}=\begin{pmatrix}\boldsymbol{v}_{\mathrm{I}}\\ \boldsymbol{v}_{\mathrm{II}}\end{pmatrix},\quad \boldsymbol{u}=\begin{pmatrix}\boldsymbol{u}_{\mathrm{I}}\\ \boldsymbol{u}_{\mathrm{II}}\end{pmatrix},\quad \boldsymbol{F}=\begin{pmatrix}\boldsymbol{F}_{\mathrm{I}}\\ \boldsymbol{F}_{\mathrm{II}}\end{pmatrix},$$

其中 $\boldsymbol{K}_{11}$ 是 $l\times l$ 的方阵，$\boldsymbol{K}_{22}$ 是 $(NP-l)\times(NP-l)$ 的方阵，$\boldsymbol{v}_{\mathrm{I}}$ 和 $\boldsymbol{u}_{\mathrm{I}}$，$\boldsymbol{F}_{\mathrm{I}}$ 都是 $l\times 1$ 的矩阵，且 $\boldsymbol{v}_{\mathrm{I}}$ 和 $\boldsymbol{u}_{\mathrm{I}}$ 的元素均为 0；$\boldsymbol{v}_{\mathrm{II}}$ 和 $\boldsymbol{u}_{\mathrm{II}}$，$\boldsymbol{F}_{\mathrm{II}}$

都是$(NP-l)\times 1$的矩阵. 因此，离散问题(12.4.19)又可写成：

求 $\boldsymbol{u}_{\mathrm{II}}=(u_{l+1},u_{l+2},\cdots,u_{NP})^{\mathrm{T}}$,

使得
$$\boldsymbol{v}_{\mathrm{II}}^{\mathrm{T}}[\boldsymbol{K}_{22}\boldsymbol{u}_{\mathrm{II}}-(\boldsymbol{F}_{\mathrm{II}}-\boldsymbol{K}_{21}\boldsymbol{u}_{\mathrm{I}})]=\boldsymbol{0} \tag{12.4.20}$$
对任意的$\boldsymbol{v}_{\mathrm{II}}=(v_{l+1},v_{l+2},\cdots,v_{NP})^{\mathrm{T}}$成立.

由于$\boldsymbol{v}_{\mathrm{II}}$的任意性，离散问题(12.4.20)化为求解方程组
$$\boldsymbol{K}_{22}\boldsymbol{u}_{\mathrm{II}}=\boldsymbol{F}_{\mathrm{II}}-\boldsymbol{K}_{21}\boldsymbol{u}_{\mathrm{I}}, \tag{12.4.21}$$
其中系数矩阵$\boldsymbol{K}_{22}$是从总刚度矩阵$\boldsymbol{K}$中划去前l行l列而得. 若$\partial\Omega_1$上的节点不是排在前l个，则在$\boldsymbol{K}$中划去相应的l行l列. $\boldsymbol{F}_{\mathrm{II}}$也类似地从$\boldsymbol{F}$得到. 在齐次边界条件(12.4.2)下，$\boldsymbol{u}_{\mathrm{I}}$应为零向量. 若边值问题在$\partial\Omega_1$是非齐次的约束边界条件，则$\boldsymbol{u}_{\mathrm{I}}$是已知的非零向量.

解离散问题(12.4.19)的另一种方法是保持刚度矩阵$\boldsymbol{K}$的阶数，将问题化为解方程组
$$\begin{pmatrix}\boldsymbol{I}_l & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{K}_{22}\end{pmatrix}\begin{pmatrix}\boldsymbol{u}_{\mathrm{I}}\\ \boldsymbol{u}_{\mathrm{II}}\end{pmatrix}=\begin{pmatrix}\boldsymbol{u}_{\mathrm{I}}^0\\ \boldsymbol{F}_{\mathrm{II}}-\boldsymbol{K}_{21}\quad\boldsymbol{u}_{\mathrm{I}}^0\end{pmatrix}, \tag{12.4.22}$$
其中$\boldsymbol{I}_l$是l阶单位阵，$\boldsymbol{u}_{\mathrm{I}}^0$是由$\partial\Omega_1$上节点$u$的值组成的向量. 它和方程组(12.4.21)是等价的.

也可以选择一个大数N，例如$N=10^{10}$，求解方程组
$$\begin{pmatrix}\widetilde{\boldsymbol{K}}_{11} & \boldsymbol{K}_{12}\\ \boldsymbol{K}_{21} & \boldsymbol{K}_{22}\end{pmatrix}\begin{pmatrix}\boldsymbol{u}_{\mathrm{I}}\\ \boldsymbol{u}_{\mathrm{II}}\end{pmatrix}=\begin{pmatrix}\widetilde{\boldsymbol{F}}_{\mathrm{I}}\\ \boldsymbol{F}_{\mathrm{II}}\end{pmatrix}, \tag{12.4.23}$$
其中分块的方法仍同上，$\widetilde{\boldsymbol{K}}_{11}$是在$\boldsymbol{K}_{11}$的对角线元素乘$N$，其他元素不变所得的矩阵，即
$$\widetilde{\boldsymbol{K}}_{11}=\begin{pmatrix}k_{11}N & k_{12} & \cdots & k_{1l}\\ k_{21} & k_{22}N & \cdots & k_{2l}\\ \vdots & \vdots & & \vdots\\ k_{l1} & k_{l2} & \cdots & k_{ll}N\end{pmatrix},$$
$\boldsymbol{K}_{12},\boldsymbol{K}_{22},\boldsymbol{K}_{21},\boldsymbol{u}_{\mathrm{I}},\boldsymbol{u}_{\mathrm{II}},\boldsymbol{F}_{\mathrm{II}}$仍同上，而

$$\widetilde{\boldsymbol{F}}_{\mathrm{I}} = \begin{pmatrix} u_1^0 k_{11} N \\ u_2^0 k_{22} N \\ \vdots \\ u_l^0 k_{ll} N \end{pmatrix},$$

其中 $u_1^0, u_2^0, \cdots, u_l^0$，为 u 在 $\partial \Omega_1$ 上节点的已知值. 解方程组(12.4.23)的实际效果和解方程组(12.4.21)的效果相同，且系数矩阵仍保持对称性.

例 12.4.3 考虑矩形域上无热源的定常温度场，边界上给出绝热条件或给定温度值，其边值问题为

$$\begin{cases} \dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} = 0 \quad (0 < x < 2, 0 < y < 2), \\ u = 50 \quad (y = 0), \quad u = 100 \quad (y = 2), \\ \dfrac{\partial u}{\partial x} = 0 \quad (x = 0), \quad \dfrac{\partial u}{\partial x} = 0 \quad (x = 2). \end{cases}$$

用线性元方法求其温度分布.

解 对区域 $\{(x, y) \mid 0 \leqslant x \leqslant 2, 0 \leqslant y \leqslant 2\}$ 做三角形剖分，共 16 个三角形单元，15 个节点. 单元编号和节点编号如图 12.20 所示.

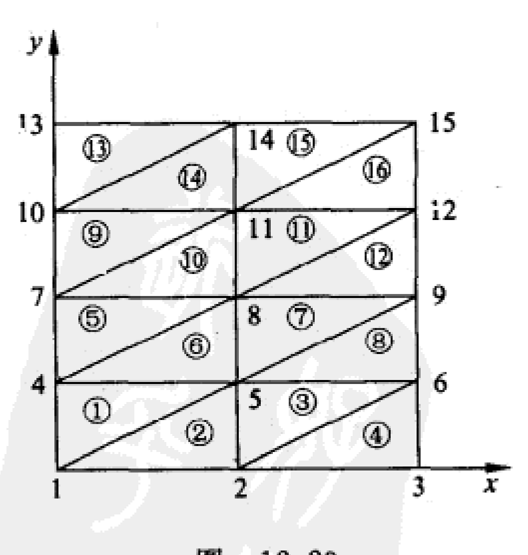


图 12.20

节点的坐标如表 12.7 所示.

表 12.7

节点	x	y	节点	x	y	节点	x	y
1	0	0	6	2	0.5	11	1	1.5
2	1	0	7	0	1	12	2	1.5
3	2	0	8	1	1	13	0	2
4	0	0.5	9	2	1	14	1	2
5	1	0.5	10	0	1.5	15	2	2

每个单元的参数 a_i, a_j, a_m 及 b_i, b_j, b_m 如表 12.8 所示.

表 12.8

单元	节点号			参数					
	i	j	m	a_i	a_j	a_m	b_i	b_j	b_m
1	4	1	5	−0.5	0	0.5	1	−1	0
2	2	5	1	0.5	0	−0.5	−1	1	0
3	5	2	6	−0.5	0	0.5	1	−1	0
4	3	6	2	0.5	0	−0.5	−1	1	0
5	7	4	8	−0.5	0	0.5	1	−1	0
6	5	8	4	0.5	0	−0.5	−1	1	0
7	8	5	9	−0.5	0	0.5	1	−1	0
8	6	9	5	0.5	0	−0.5	−1	1	0
9	10	7	11	−0.5	0	0.5	1	−1	0
10	8	11	7	0.5	0	−0.5	−1	1	0
11	11	8	12	−0.5	0	0.5	1	−1	0
12	9	12	8	0.5	0	−0.5	−1	1	0
13	13	10	14	−0.5	0	0.5	1	−1	0
14	11	14	10	0.5	0	−0.5	−1	1	0
15	14	11	15	−0.5	0	0.5	1	−1	0
16	12	15	11	0.5	0	−0.5	−1	1	0

$$
\frac{1}{4}\begin{bmatrix}
5 & -1 & 0 & -4 & & & & & & & & & & & \\
-1 & 10 & -1 & 0 & -8 & & & & & & & & & & \\
0 & -1 & 5 & 0 & 0 & -4 & & & & & & & & & \\
-4 & 0 & 0 & 10 & -2 & 0 & -4 & & & & & & & & \\
 & -8 & 0 & -2 & 20 & -2 & 0 & -8 & & & & & & & \\
 & & -4 & 0 & -2 & 10 & 0 & 0 & -4 & & & & & & \\
 & & & -4 & 0 & 0 & 10 & -2 & 0 & -4 & & & & & \\
 & & & & -8 & 0 & -2 & 20 & -2 & 0 & -8 & & & & \\
 & & & & & -4 & 0 & -2 & 10 & 0 & 0 & -4 & & & \\
 & & & & & & -4 & 0 & 0 & 10 & -2 & 0 & -4 & & \\
 & & & & & & & -8 & 0 & -2 & 20 & -2 & 0 & -8 & \\
 & & & & & & & & -4 & 0 & -2 & 10 & 0 & 0 & -4 \\
 & & & & & & & & & -4 & 0 & 0 & 5 & -1 & 0 \\
 & & & & & & & & & & -8 & 0 & -1 & 10 & -1 \\
 & & & & & & & & & & & -4 & 0 & -1 & 5
\end{bmatrix}
$$

进行边界约束条件处理时，注意这类边界上节点编号是 1，2，3 和 13，14，15 共 6 个点. 用方程(12.4.22)的方法，有

$$
u_1^0 = u_2^0 = u_3^0 = 50, \quad u_{13}^0 = u_{14}^0 = u_{15}^0 = 100.
$$

把总刚度矩阵中第 1，2，3，13，14，15 行和列的对角线元素改为 1，其他元素改为 0，方程组的右端也按式(12.4.22)计算，最后得到方程组

$$
\frac{1}{4}\begin{bmatrix}
4 & & & & & & & & & & & & & & \\
 & 4 & & & & & & & & & & & & & \\
 & & 4 & & & & & & & & & & & & \\
 & & & 10 & -2 & 0 & -4 & & & & & & & & \\
 & & & -2 & 20 & -2 & 0 & -8 & & & & & & & \\
 & & & 0 & -2 & 10 & 0 & 0 & -4 & & & & & & \\
 & & & -4 & 0 & 0 & 10 & -2 & 0 & -4 & & & & & \\
 & & & & -8 & 0 & -2 & 20 & -2 & 0 & -8 & & & & \\
 & & & & & -4 & 0 & -2 & 10 & 0 & 0 & -4 & & & \\
 & & & & & & -4 & 0 & 0 & 10 & -2 & 0 & & & \\
 & & & & & & & -8 & 0 & -2 & 20 & -2 & & & \\
 & & & & & & & & -4 & 0 & -2 & 10 & & & \\
 & & & & & & & & & & & & 4 & & \\
 & & & & & & & & & & & & & 4 & \\
 & & & & & & & & & & & & & & 4
\end{bmatrix}
\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \\ u_{10} \\ u_{11} \\ u_{12} \\ u_{13} \\ u_{14} \\ u_{15} \end{bmatrix}
=
\begin{bmatrix} 50 \\ 50 \\ 50 \\ 50 \\ 100 \\ 50 \\ 0 \\ 0 \\ 0 \\ 100 \\ 200 \\ 100 \\ 100 \\ 100 \\ 100 \end{bmatrix}.
$$

按照单元刚度矩阵公式计算时，由于本例边界条件中 $\sigma=g=0$，所以变分问题中不出现 γ_n 上的线积分项，即 $\widetilde{K}_{e_n}$ 为零矩阵. 由单元参数表可看到，所有编号为奇数的单元和编号为偶数的单元的参数分别都相同，三角形元的面积亦相同. 所以编号为奇数的单元的单元刚度矩阵都是一样的，编号为偶数的单元也同样. 可计算得

$$\boldsymbol{K}_{e_1}=\boldsymbol{K}_{e_2}=\frac{1}{4}\begin{pmatrix}5 & -4 & -1\\ -4 & 4 & 0\\ -1 & 0 & 1\end{pmatrix}.$$

第 1 号单元的节点 P_i,P_j,P_m 编号为第 4，第 1，第 5 号节点，单元刚度矩阵"扩大"后得

$$\frac{1}{4}\begin{pmatrix}4 & \cdots & -4 & 0 & \\ \vdots & & \vdots & \vdots & \\ -4 & \cdots & 5 & -1 & \\ 0 & \cdots & -1 & 1 & \\ & & & & \ddots\end{pmatrix}.$$

同理，第 2，第 3 号单元的刚度矩阵"扩大"后得

$$\frac{1}{4}\begin{pmatrix}1 & -5 & \cdots & 0 & \\ -1 & 5 & \cdots & -4 & \\ \vdots & \vdots & & \vdots & \\ 0 & -4 & \cdots & 4 & \\ & & & & \ddots\end{pmatrix},$$

$$\frac{1}{4}\begin{pmatrix} & \vdots & & \vdots & \vdots & \\ \cdots & 4 & \cdots & -4 & 0 & \\ & \vdots & & \vdots & \vdots & \\ \cdots & -4 & \cdots & 5 & -1 & \\ \cdots & 0 & \cdots & -1 & 1 & \\ & & & & & \ddots\end{pmatrix}.$$

其他单元刚度矩阵类似地"扩大"，叠加起来得到总刚度矩阵

该方程组的解是

$$u_1 = u_2 = u_3 = 50, \qquad u_4 = u_5 = u_6 = 62.5$$
$$u_7 = u_8 = u_9 = 75, \qquad u_{10} = u_{11} = u_{12} = 87.5,$$
$$u_{13} = u_{14} = u_{15} = 100.$$

这就是线性元方法求出的边值问题的解在节点上的值.

12.4.3 三角形单元上的高次插值

三角形单元除了用上述线性多项式插值外，还可以用变量 x，y 的 k 次多项式插值. k 次完全多项式有$\frac{1}{2}(k+1)(k+2)$项，即

$$\begin{array}{ccccccc} & & & 1 & & & \quad 0\text{ 次} \\ & & x & & y & & \quad 1\text{ 次} \\ & x^2 & & xy & & y^2 & \quad 2\text{ 次} \\ x^3 & & x^2y & & xy^2 & \quad y^3 & \quad 3\text{ 次} \\ & & & \vdots & & & \quad \vdots \end{array}$$

所以 k 次拉格朗日插值需要$\frac{1}{2}(k+1)(k+2)$个节点，一般按照三角形各边的 k 等分点及它们连线的交点作为节点，如图 12.21 所示.

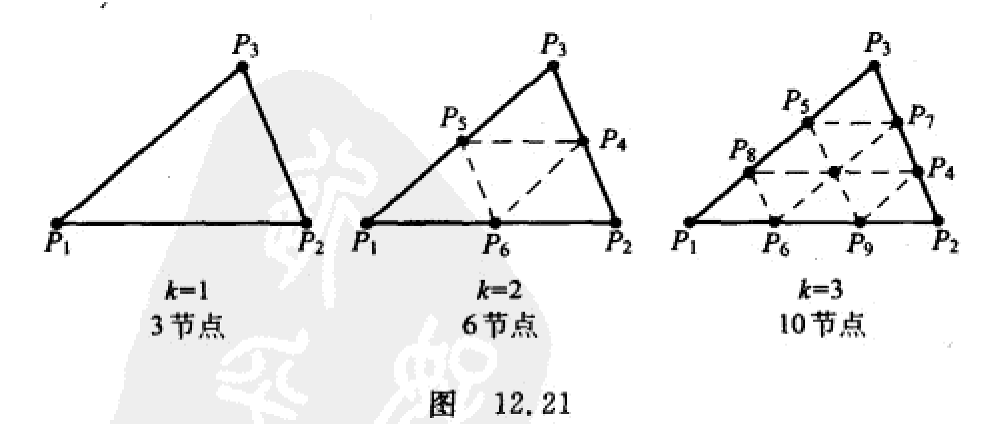


图 12.21

设三角形元 $e=\Delta P_1P_2P_3$，其面积是 Δ_e，节点 P_i 的坐标是 (x_i, y_i). 对于 e 内任意一点 $P(x,y)$，定义它的**面积坐标**(area

coordinate)为

$$L_1=\frac{1}{2\Delta_e}\begin{vmatrix}x & y & 1\\ x_2 & y_2 & 1\\ x_3 & y_3 & 1\end{vmatrix},$$

$$L_2=\frac{1}{2\Delta_e}\begin{vmatrix}x & y & 1\\ x_3 & y_3 & 1\\ x_1 & y_1 & 1\end{vmatrix}, \qquad (12.4.24)$$

$$L_3=\frac{1}{2\Delta_e}\begin{vmatrix}x & y & 1\\ x_1 & y_1 & 1\\ x_2 & y_2 & 1\end{vmatrix}.$$

这里的 L_1,L_2,L_3 即式(12.4.13)的 N_i,N_j,N_m. 由式(12.4.14)和式(12.4.15)有

$$L_i(P_j)=\begin{cases}0, & i\neq j,\\ 1, & i=j,\end{cases}\quad (i,j=1,2,3).$$

$$\sum_{i=1}^{3}L_i=1,\quad \sum_{i=1}^{3}L_ix_i=x,\quad \sum_{i=1}^{3}L_iy_i=y.$$

点 $P(x,y)$ 的面积坐标 L_1,L_2,L_3 的几何意义用图 12.22 所示的三角形面积比表示.

$$L_1=\frac{S_{\Delta_{PP_2P_3}}}{S_{\Delta_{P_1P_2P_3}}},\quad L_2=\frac{S_{\Delta_{PP_3P_1}}}{S_{\Delta_{P_1P_2P_3}}},$$

$$L_3=\frac{S_{\Delta_{PP_1P_2}}}{S_{\Delta_{P_1P_2P_3}}}.$$

图 12.22

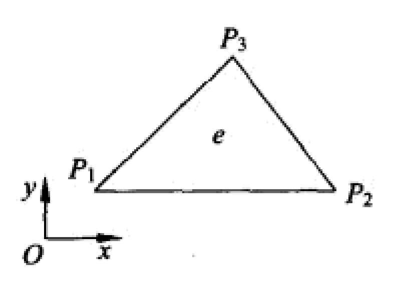
作 x-y 平面到 ξ-η 平面的变换

$$\begin{cases}\xi=L_1(x,y),\\ \eta=L_2(x,y).\end{cases}\qquad (12.4.25)$$

其逆变换为:

$$\begin{cases} x = (x_1 - x_3)\xi + (x_2 - x_3)\eta + x_3, \\ y = (y_1 - y_3)\xi + (y_2 - y_3)\eta + y_3. \end{cases} \tag{12.4.26}$$

变换(12.4.25)将 x-y 平面上的三角形元 e 变换为 ξ-η 平面上的标准三角形 $\hat{e}$，如图 12.23 所示.



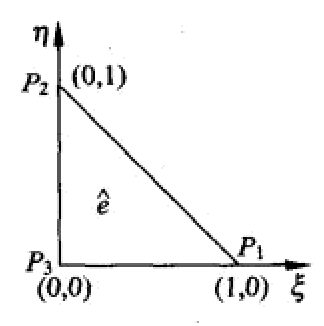


图 12.23

在近似变分问题中，积分的计算可以在 $\hat{e}$ 上进行，积分公式为

$$\begin{aligned} &\iint_e F(L_1(x,y), L_2(x,y), L_3(x,y))\,\mathrm{d}x\mathrm{d}y \\ &= 2\Delta_e \int_0^1 \mathrm{d}\xi \int_0^{1-\xi} F(\xi, \eta, 1-\xi-\eta)\,\mathrm{d}\eta. \end{aligned} \tag{12.4.27}$$

其中一个重要的例子是

$$\iint_e L_1^{\lambda_1} L_2^{\lambda_2} L_3^{\lambda_3}\,\mathrm{d}x\mathrm{d}y = \frac{\lambda_1!\lambda_2!\lambda_3!}{(\lambda_1+\lambda_2+\lambda_3+2)!} \cdot 2\Delta_e, \tag{12.4.28}$$

式中 $\lambda_1, \lambda_2, \lambda_3$ 为非负整数.

对三角形单元，经常用到数值积分方法，表 12.9 列举了一些数值积分公式，一般形式是

$$\iint_e F(L_1, L_2, L_3)\,\mathrm{d}x\mathrm{d}y \approx \Delta_e \sum_{k=1}^{m} \rho^{(k)} F(L_1^{(k)}, L_2^{(k)}, L_3^{(k)}), \tag{12.4.29}$$

其中 m 是积分节点的个数，$\rho^{(k)}$ 是对应第 k 个节点的权系数，$L_1^{(k)}$，

$L_2^{(k)}, L_3^{(k)}$ 是第 k 个节点的面积坐标，表中的精度次数 n 是指公式(12.4.29)对不超过 n 次的多项式是准确的.

表 12.9

节点个数 m	节点坐标(L_1, L_2, L_3)	权数 ρ	精度次数 n
1	$\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$	1	1
3	$\left(\frac{1}{2}, \frac{1}{2}, 0,\right)$ $\left(0, \frac{1}{2}, \frac{1}{2}\right)$ $\left(\frac{1}{2}, 0, \frac{1}{2}\right)$	$\frac{1}{3}$ $\frac{1}{3}$ $\frac{1}{3}$	2
4	$\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$ $\left.\begin{matrix}(0.6, 0.2, 0.2)\\(0.2, 0.6, 0.2)\\(0.2, 0.2, 0.6)\end{matrix}\right\}$	$-\frac{27}{48}$ $\frac{25}{48}$	3
7 $\alpha_1 = 0.0597158717$ $\beta_1 = 0.4701420641$ $\alpha_2 = 0.7974269853$ $\beta_2 = 0.1012865073$	$\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$ $\left.\begin{matrix}(\alpha_1, \beta_1, \beta_1)\\(\beta_1, \alpha_1, \beta_1)\\(\beta_1, \beta_1, \alpha_1)\end{matrix}\right\}$ $\left.\begin{matrix}(\alpha_2, \beta_2, \beta_2)\\(\beta_2, \alpha_2, \beta_2)\\(\beta_2, \beta_2, \alpha_2)\end{matrix}\right\}$	0.2250000000 0.1323941527 0.1259391805	5

在三角形单元 e 上，如果是 k 次拉格朗日插值中 $k=2$ 的情形，则有 6 个节点，节点 P_i 的面积坐标分别是

$$(1,0,0), \quad (0,1,0), \quad (0,0,1),$$
$$\left(0,\frac{1}{2},\frac{1}{2}\right), \quad \left(\frac{1}{2},0,\frac{1}{2}\right), \quad \left(\frac{1}{2},\frac{1}{2},0\right).$$

若函数 u 在节点 P_i 上的值为 $u_i (i=1,2,\cdots,6)$，则在 e 上 u 的二次插值函数是

$$u_h = \sum_{i=1}^{6} u_i N_i, \qquad (12.4.30)$$

其中 N_i 为三角形单元上的**二次插值基函数**(basis function of quadrtic interpolation):

$$N_i = L_i(2L_i - 1) \quad (i = 1,2,3)$$

$$N_4 = 4L_2L_3, \quad N_5 = 4L_3L_1, \quad N_6 = 4L_1L_2. \qquad (12.4.31)$$

由式(12.4.30)和式(12.4.31),可以类似于线性插值情形,得到近似变分问题和对应的离散问题.

如果是 $k=3$ 的情形,则 10 个节点的面积坐标分别是

$(1,0,0)$, $(0,1,0)$, $(0,0,1)$,

$\left(0,\frac{2}{3},\frac{1}{3}\right)$, $\left(\frac{1}{3},0,\frac{2}{3}\right)$, $\left(\frac{2}{3},\frac{1}{3},0\right)$,

$\left(0,\frac{1}{3},\frac{2}{3}\right)$, $\left(\frac{2}{3},0,\frac{1}{3}\right)$, $\left(\frac{1}{3},\frac{2}{3},0\right)$, $\left(\frac{1}{3},\frac{1}{3},\frac{1}{3}\right)$.

三角形单元上的**三次插值基函数**(basis functions of cubic interpolation)是

$$\begin{aligned}
N_i &= \frac{1}{2}L_i(3L_i - 1)(3L_i - 2) \quad (i = 1,2,3),\\
N_4 &= \frac{9}{2}L_2L_3(3L_2 - 1),\\
N_5 &= \frac{9}{2}L_3L_1(3L_3 - 1),\\
N_6 &= \frac{9}{2}L_1L_2(3L_1 - 1),\\
N_7 &= \frac{9}{2}L_2L_3(3L_3 - 1),\\
N_8 &= \frac{9}{2}L_3L_1(3L_1 - 1),\\
N_9 &= \frac{9}{2}L_1L_2(3L_2 - 1),\\
N_{10} &= 27L_1L_2L_3.
\end{aligned} \qquad (12.4.32)$$

在三角形单元 e 上,函数 u 的三次插值函数表示为

$$u_h = \sum_{i=1}^{10} u_i N_i.$$

在做了三角形剖分的区域 Ω 上，得到分片三次插值函数 $u_h(x,y)$，$(x,y)\in\Omega$，在某一个单元的一条边上，$u_h(x,y)$可以表示为该边弧长参数 t 的三次函数，可由该边上 4 个节点的 u_i 值惟一确定，所以 $u_h(x,y)$在每条边上都是连续的，有 $u_h\in C(\bar{\Omega})$. 分片线性插值和分片二次插值也有此性质.

12.4.4 矩形单元

除了三角形剖分外，矩形剖分也是一种应用广泛的剖分方法，矩形剖分方法把边值问题(12.4.1)～(12.4.3)的求解区域 Ω 剖分为若干个矩形单元(rectangular finite element)的组合，这些矩形单元的边都平行于坐标轴，如图 12.24 所示，其中(x_c,y_c)是矩形的形心.

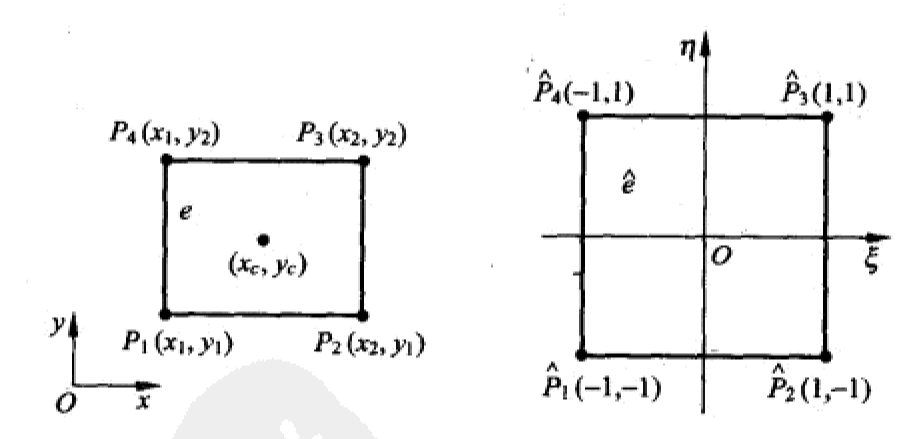


图 12.24

矩阵单元 e 中任一点 $P(x,y)$的**局部坐标**(local coordinate)定义为

$$\begin{cases} \xi = \dfrac{2(x-x_c)}{x_2-x_1}, \\ \eta = \dfrac{2(y-y_c)}{y_2-y_1}. \end{cases} \tag{12.4.33}$$

式(12.4.33)可以看成 x-y 平面至 ξ-η 平面的一个变换,它将单元 e 变换到标准单元 $\hat{e}$,如图 12.24 所示.

矩形单元上双线性插值函数共有 4 项,即

$$u_h(x,y) = a + bx + cy + dxy. \qquad (12.4.34)$$

在 e 上,4 个节点的函数值可以确定 4 个系数 a,b,c,d,而 $1,x,y$,xy 这 4 项可以看成

$$\begin{pmatrix}1\\x\end{pmatrix}(1,\quad y) = \begin{pmatrix}1 & y\\ x & xy\end{pmatrix}.$$

在式(12.4.34)中,若 x 或 y 有一个变量固定,u_h 就是另一变量的线性函数.

在标准单元 $\hat{e}$ 上,节点 $\hat{P}_i$ 坐标为 $(\xi_i,\eta_i)(i=1,2,3,4)$,如图 12.24 所示.设在 $\hat{P}_i$ 点 u 的值为 $u_i(i=1,2,3,4)$,则在 $\hat{e}$ 上的双线性插值函数为

$$u_h = \sum_{i=1}^{4} u_i N_i, \qquad (12.4.35)$$

其中 $N_i(i=1,2,3,4)$ 为**双线性插值基函数**(basis functions of bilinear interpolation):

$$N_i = \frac{1}{4}(1+\xi_i\xi)(1+\eta_i\eta) \quad (i=1,2,3,4). \qquad (12.4.36)$$

矩形单元上双二次插值共有 9 项,可看成

$$\begin{pmatrix}1\\x\\x^2\end{pmatrix}(1\quad y\quad y^2) = \begin{pmatrix}1 & y & y^2\\ x & xy & xy^2\\ x^2 & x^2y & x^2y^2\end{pmatrix}.$$

双二次插值共需 9 个节点,如图 12.25 所示.

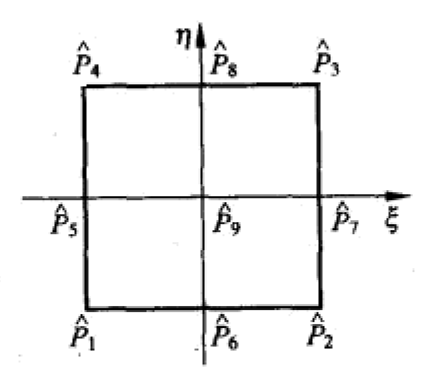


图 12.25

双二次插值基函数(basic functions of biquadratic interpolation)为

$$
\begin{cases}
N_1 = \frac{1}{4}\xi\eta(\xi - 1)(\eta - 1), \\
N_2 = \frac{1}{4}\xi\eta(\xi + 1)(\eta - 1), \\
N_3 = \frac{1}{4}\xi\eta(\xi + 1)(\eta + 1), \\
N_4 = \frac{1}{4}\xi\eta(\xi - 1)(\eta + 1), \\
N_5 = \frac{1}{2}\xi(\xi - 1)(1 - \eta^2), \\
N_6 = \frac{1}{2}\eta(\eta - 1)(1 - \xi^2), \\
N_7 = \frac{1}{2}\xi(\xi + 1)(1 - \eta^2), \\
N_8 = \frac{1}{2}\eta(\eta + 1)(1 - \xi^2), \\
N_9 = (1 - \xi^2)(1 - \eta^2).
\end{cases}
\tag{12.4.37}
$$

在双二次插值式中去掉 x^2y^2 项,对应在 $\hat{e}$ 中去掉内部节点 $\hat{P}_9$,得到**不完全的双二次插值**,不完全双二次插值基函数是

$$
\begin{cases}
N_1 = -\frac{1}{4}(1 - \xi)(1 - \eta)(1 + \xi + \eta), \\
N_2 = -\frac{1}{4}(1 + \xi)(1 - \eta)(1 - \xi + \eta), \\
N_3 = -\frac{1}{4}(1 + \xi)(1 + \eta)(1 - \xi - \eta), \\
N_4 = -\frac{1}{4}(1 - \xi)(1 + \eta)(1 + \xi - \eta), \\
N_5 = \frac{1}{2}(1 - \eta^2)(1 - \xi), \\
N_6 = \frac{1}{2}(1 - \xi^2)(1 - \eta), \\
N_7 = \frac{1}{2}(1 - \eta^2)(1 + \xi), \\
N_8 = \frac{1}{2}(1 - \xi^2)(1 + \eta).
\end{cases}
\tag{12.4.38}
$$

在 Ω 上分片线性插值、分片二次插值和分片不完全二次插值所得的分片插值函数，都属于 $C(\bar{\Omega})$.

12.4.5 等参数单元

设 $N_i(i=1,2,3,4)$ 是式(12.4.36)所示的双线性插值基函数，作变换

$$\begin{cases} x = \sum_{i=1}^{4} x_i N_i(\xi,\eta), \\ y = \sum_{i=1}^{4} y_i N_i(\xi,\eta). \end{cases} \tag{12.4.39}$$

此变换将 x-y 平面的任意四边形单元 e 变换到 ξ-η 平面的标准正方形单元 $\hat{e}$，如图 12.26 所示.

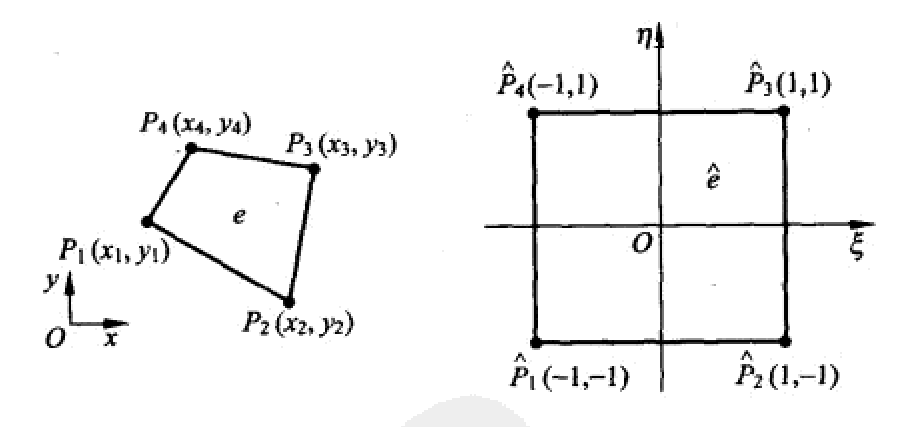


图 12.26

在 $\hat{e}$ 上取插值公式为

$$u_h = \sum_{i=1}^{4} u_i N_i(\xi,\eta), \tag{12.4.40}$$

即可做近似变分问题计算，这种任意四边形单元的方法称为 4 节点四边形等参数单元(isoparametric element)方法.

设 $N_i(i=1,2,\cdots,8)$ 为式(12.4.38)的不完全双二次插值基函数，则 8 节点四边形等参数单元的插值公式为

$$u_h = \sum_{i=1}^{8} u_i N_i(\xi,\eta). \tag{12.4.41}$$

对应的变换为

$$\begin{cases} x = \sum_{i=1}^{8} x_i N_i(\xi,\eta), \\ y = \sum_{i=1}^{8} y_i N_i(\xi,\eta). \end{cases} \tag{12.4.42}$$

此变换将图 12.27 的单元 e 变换到标准单元 $\hat{e}$，其中 e 的边界都是二次曲线.

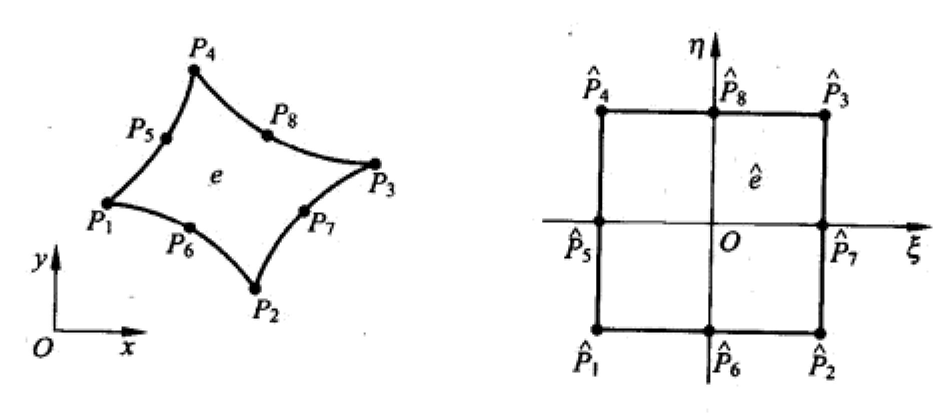


图 12.27

13 多重网格法

多重网格法(multigrid method)是由前苏联人 Fedorenk 在 1964 年提出的. 20 世纪 70 年代,A. Brandt 和 W. Hackbush 重新认识多重网格法的效率. 80 年代以后,多重网格学科的基本核心,包括基本原则、传统应用及理论都已建立,出现了一系列的多重网格法的文章和专著,其中有代表性的是参考文献[59～66].

多重网格法是一种"近乎最优"的特殊的迭代途径. 在偏微分方程数值解中,当离散网格步长 h 变小时,它的收敛速度并不减慢,而经典的迭代法随 h 的变小而变慢. 因此,多重网格法达到问题预定精度所需的计算工作量仅与未知量的个数 N 成正比,即 $O(N)$,比一般迭代法有效得多.

多重网格法广泛应用于微分方程和积分方程的数值解,特别是椭圆型偏微分方程. 除此之外,还可应用于求解抛物型问题和其他依赖于时间的问题、本征值问题、分歧问题、非线性问题和直接用于无几何意义的代数方程组(称为代数多重网格法);在向量机或并行计算机上使用多重网格法更有效. 因此,多重网格法是一种通用的非常有效的特殊类型的迭代法,已相当成功地应用于流体计算、结构力学计算和高度非线性的半导体器件模拟计算等领域.

本章侧重于介绍多重网格法的基本原理、基本要素、多重网格算法,有关收敛性结论和计算机上的执行性能.

13.1 多重网格法基本原理

13.1.1 模型

要求解的连续问题是

$$Lu = f, \tag{13.1.1}$$

其中 L 是一个微分算子或积分算子或泛函求极值算子.

例 13.1.1 两点边值问题:

$$\begin{cases} -u''(x)+\sigma u(x) = f(x), & 0\leqslant x\leqslant 1, \sigma\geqslant 0, \\ u(0)=u(1)=0, \end{cases} \tag{13.1.2}$$

则 $L=-\dfrac{\mathrm{d}^2}{\mathrm{d}x^2}+\sigma$.

例 13.1.2 二维泊松方程第一边值问题:

$$\begin{cases} -(u_{xx}+u_{yy}) = f(x,y), \\ (x,y)\in\Omega=\{(x,y)\mid 0<x,y<1\} \\ u=0, \quad (x,y)\in\partial\Omega \end{cases} \tag{13.1.3}$$

则 $L=-\left(\dfrac{\partial^2}{\partial x^2}+\dfrac{\partial^2}{\partial y^2}\right)$.

不论用有限差分法还是用有限元法求解这两个问题,都是首先对解域进行剖分,然后对微分算子进行离散,得到离散后的方程组

$$\boldsymbol{L}_h\boldsymbol{u}_h = \boldsymbol{f}_h. \tag{13.1.4}$$

对例 13.1.1 的解域[0,1],均匀剖分成 N 等分,步长 $h=1/N$,对微分算子用中心差分离散,得到如下的离散方程组:

$$\begin{cases} \dfrac{-u_{j-1}+2u_j-u_{j+1}}{h^2}+\sigma u_j = f(x_j), & 1\leqslant j\leqslant N-1, \\ u_0=u_N=0, \end{cases} \tag{13.1.5}$$

则有

$$L_h = \frac{1}{h^2}\begin{pmatrix} 2+\sigma h^2 & -1 & & & \\ -1 & 2+\sigma h^2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2+\sigma h^2 & -1 \\ & & & -1 & 2+\sigma h^2 \end{pmatrix}, \tag{13.1.6}$$

其中 $L_h \in \mathbb{R}^{(N-1)\times(N-1)}$，$u_h$，$f_h \in \mathbb{R}^{N-1}$.

对例 13.1.2，若其解域为方形，I 和 J 分别为 x 方向和 y 方向的等分数，则 $h_x = 1/I$，$h_y = 1/J$. 用 Ω_h 表示该二维网格点的集合，$u(x_i, y_j)$ 表示微分方程的精确解，u_{ij} 表示差分方程精确解. 用五点差分格式去离散微分方程(13.1.3)，得到差分方程组

$$\begin{cases} \dfrac{-u_{i-1,j}+2u_{ij}-u_{i+1,j}}{h_x^2}+\dfrac{-u_{i,j-1}+2u_{ij}-u_{i,j+1}}{h_y^2}=f_{ij}, \\ \qquad\qquad 1\leqslant i\leqslant I-1,\quad 1\leqslant j\leqslant J-1, \\ u_{ij}=0,\quad i=0,I\quad (0\leqslant j\leqslant J),\quad j=0,J,\quad 0\leqslant i\leqslant I. \end{cases} \tag{13.1.7}$$

若 $h_x = h_y = h$，按字典顺序排列 $(I-1)\times(J-1)$ 个未知量，得

$$L_h = \frac{1}{h^2}\begin{pmatrix} A & -I & & & \\ -I & A & -I & & \\ & \ddots & \ddots & \ddots & \\ & & -I & A & -I \\ & & & -I & A \end{pmatrix}. \tag{13.1.8}$$

其中 $A, I \in \mathbb{R}^{(I-1)\times(I-1)}$，$A$ 为严格对角占优的三对角矩阵，I 为单位矩阵，L_h 为 $(I-1)\times(J-1)$ 阶的大型稀疏矩阵.

13.1.2 多重网格法思想

对于离散后得到的线性代数方程组(13.1.4)，传统的迭代解法是在确定的一种网格 Ω_h 上采用某种迭代解法. 例如雅可比迭

代法、高斯-赛德尔迭代法(简称 G-S 法)、SOR 法、交替方向法等以及它们的改进技术.而多重网格法恰恰不是在一种确定的网格上求解,而采用不同等级的均匀网格剖分.

假定有一组网格 $\Omega_{h_0},\Omega_{h_1},\cdots,\Omega_{h_M}$,称为网格序列 $\Omega_k(k=0,1,\cdots,M)$.随着 k 的增加,差分网格愈来愈细,这些网格都用同一种方式剖分得到,都逼近同一个区域 Ω.我们感兴趣的是要求解 Ω_M 上的差分方程组

$$\boldsymbol{L}_M\boldsymbol{u}_M=\boldsymbol{f}_M. \tag{13.1.9}$$

称在 Ω_k 上解 $\boldsymbol{L}_k\boldsymbol{u}_k=\boldsymbol{f}_k$ 的问题为 Ω_k 问题,称在 Ω_M 上求解差分方程(13.1.9)的问题为 Ω_M 问题.

对于线性问题,令 $\bar{\boldsymbol{u}}_M$ 为 Ω_M 问题的近似解,$\boldsymbol{u}_M$ 为精确解,则有

$$\boldsymbol{u}_M=\bar{\boldsymbol{u}}_M+\boldsymbol{v}_M,$$

其中$\boldsymbol{v}_M$ 称为**修正量**.代入方程(13.1.9)后,得

$$\boldsymbol{L}_M\boldsymbol{v}_M=\boldsymbol{d}_M.$$

称上式为**亏损方程**,其中

$$\boldsymbol{d}_M=\boldsymbol{f}_M-\boldsymbol{L}_M\bar{\boldsymbol{u}}_M$$

称为**亏损量**(defective number).

线性多重网格法是一种特殊的迭代过程,它把 Ω_M 上求解方程(13.1.9)与在粗网 Ω_k 上求解亏损方程

$$\boldsymbol{L}_k\boldsymbol{v}_k=\boldsymbol{d}_k \tag{13.1.10}$$

相结合,而求解 Ω_k 上的亏损方程又把在Ω_k 上迭代(13.1.10)与求解 Ω_{k-1}上的亏损方程相结合.这样一种巧妙的结合构成一种收敛速度很快的迭代方法,这就是多重网格法的基本思想.

13.1.3 双网格方法

双网格方法(two-grid method)是多重网格法的基础.首先叙述仅附加一种网格的双网格方法,然后再推广到一系列网格的多

重网格方法.

双网格方法用两种网格，一种用 Ω_H 表示，另一种用 Ω_h 表示. 网格步长分别为 H 和 h. 通常 $H=2h$. 在 Ω_H 和 Ω_h 上的差分算子分别为 $\boldsymbol{L}_H$ 和 $\boldsymbol{L}_h$，且设 $\boldsymbol{L}_H^{-1}$ 和 $\boldsymbol{L}_h^{-1}$ 存在.

现在的任务是在 Ω_h 上求解方程组.

$$\boldsymbol{L}_h\boldsymbol{u}_h = \boldsymbol{f}_h. \tag{13.1.11}$$

粗细两种网格上的值需要转移，下面定义由粗网到细网及由细网到粗网上网格函数的转移算子.

插值(interpolation)**算子或延拓**(prolongation)**算子** $\boldsymbol{I}_H^h$ 表示粗网到细网上值的转移. 如一维线性插值

$$\boldsymbol{I}_H^h \overset{\text{def}}{=} \frac{1}{2}]1\quad 2\quad 1[_H^h \quad 或 \quad \boldsymbol{I}_H^h \overset{\text{def}}{=} \frac{1}{2}\begin{pmatrix} 1 & & & \\ 2 & & & \\ 1 & 1 & & \\ & 2 & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \\ & & & 2 \\ & & & 1 \end{pmatrix}, \tag{13.1.12}$$

式(13.1.12)表示粗网上的值按此权系数分配到邻近点的细网上去，

$$\boldsymbol{L}_H^h\boldsymbol{u}_H = \boldsymbol{u}_h,$$

即
$$\begin{cases} u_{2j}^h = u_j^H, \\ u_{2j+1}^h = \dfrac{u_j^H + u_{j+1}^H}{2}, \end{cases} \quad 0 \leqslant j \leqslant \frac{N}{2} - 1.$$

限制(restriction)**算子** $\boldsymbol{I}_h^H$ 表示由细网到粗网上值的转移. 如一维限制算子

$$I_h^H \overset{\text{def}}{=} \frac{1}{4}[1 \quad 2 \quad 1]_h^H \quad 或 \quad I_h^H \overset{\text{def}}{=} \frac{1}{4}\begin{pmatrix} 1 & 2 & 1 & & & & \\ & & 1 & 2 & 1 & & \\ & & & & \ddots & & \\ & & & & 1 & 2 & 1 \end{pmatrix}, \tag{13.1.13}$$

$$I_h^H u_h = u_H,$$

即
$$u_j^H = \frac{u_{2j-1}^h + 2u_{2j}^h + u_{2j+1}^h}{4} \quad \left(1 \leqslant j \leqslant \frac{N}{2} - 1\right).$$

下面叙述用双网格方法求解方程(13.1.11)时，由已知 $u_h^{(n)}$ 计算 $u_h^{(n+1)}$ 的一个迭代步或一个循环步. 共分三个阶段：

(1) **前光滑**(pre-smoothing)：

给定初值 $u_h^{(n)}$，选定一种松弛法，在 Ω_h 上对方程(13.1.11)作 ν_1($\nu_1 = 1$ 或 2)次迭代(也称松弛)计算 $\bar{u}_h^{(n)}$

$$\bar{u}_h^{(n)} \overset{\text{def}}{=} S^{\nu_1}(u_h^{(n)}, L_h, f_h)$$

(2) **粗网修正**(coarse-grid correction，简称 CGC)：

① 计算细网亏损量：$d_h^{(n)} = f_h - L_h \bar{u}_h^{(n)}$；

② 限制亏损量(由细网到粗网)：$d_H^{(n)} = I_h^H d_h^{(n)}$；

③ 在 Ω_H 上求

$$L_H v_H^{(n)} = d_H^{(n)} \tag{13.1.14}$$

的精确解 $v_H^{(n)}$.

④ 插值修正量(由粗网到细网)：$v_h^{(n)} = I_H^h v_H^{(n)}$.

⑤ 对 $\bar{u}_h^{(n)}$ 作修正：$\hat{u}_h^{(n)} = \bar{u}_h^{(n)} + v_h^{(n)}$.

(3) 后光滑(post-smoothing)：

以 $\hat{u}_h^{(n)}$ 为迭代的初始近似值，在 Ω_h 上对方程(13.1.11)作 ν_2($\nu_2 = 1$ 或 2)次迭代计算 $u_h^{(n+1)}$，

$$u_h^{(n+1)} \overset{\text{def}}{=} S^{\nu_2}(\hat{u}_h^{(n)}, L_h, f_h).$$

为直观起见，用下列路径表示双网格的一个循环步的计算：

$$\begin{array}{llll}
\Omega_h & u_h^{(n)} \xrightarrow{S^{\nu_1}} \bar{u}_h^{(n)} \to d_h^{(n)} = f_h - L_h \bar{u}_h^{(n)}, & v_h^{(n)} \to \hat{u}_h^{(n)} = \bar{u}_h^{(n)} + v_h^{(n)} \xrightarrow{S^{\nu_2}} u_h^{(n+1)} \\
& \quad\quad \downarrow I_h^H & \uparrow I_H^h \\
\Omega_H & \quad\quad d_H^{(n)} \quad \longrightarrow & L_H v_H^{(n)} = d_H^{(n)}
\end{array}$$

这过程示于图 13.1，图中○、□、↘、↗分别表示松弛，求精确解和限制亏损量及插值修正量.

这样一种特殊迭代过程的迭代公式为

$$u_h^{(n+1)} = M_h^H u_h^{(n)} + N_h(f_h),$$

图 13.1

其双网格迭代矩阵为

$$M_h^H = S^{\nu_2} K_h^H S^{\nu_1}, \tag{13.1.15}$$

其中

$$K_h^H \stackrel{\text{def}}{=} I_h - I_H^h L_H^{-1} I_h^H L_h \tag{13.1.16}$$

是粗网修正的迭代矩阵.

13.1.4 多重网格法原理——一维模型问题分析

细网上松弛 S^{ν_1}，S^{ν_2} 和粗网修正是双网格方法的两大要素. 若仅在细网上作松弛，如雅可比松弛，高斯-赛德尔松弛和 SOR 松弛. 当网格步长 $h\to 0$ 时，收敛性变得很差. 若单独用粗网修正，在迭代时，粗网修正的迭代矩阵 K_h^H 的谱半径 $\rho(K_h^H)\geqslant 1$，从而不收敛. 如果把细网松弛和粗网修正相结合，则构成一种收敛速度很快的一种特殊的迭代法. 下面以 13.1.1 节中的一维模型($\sigma=0$)为例，对这两个要素作具体分析.

1. 细网松弛

细网上方程 $Au=b$ 的定常迭代法为

$$u^{(n+1)} = Bu^{(n)} + g. \tag{13.1.17a}$$

将方程的系数矩阵 A 分解为 $A=D-L-U$，其中 D 为 A 的对角线元素构成的对角矩阵，$-L$ 为 A 的下三角部分构成的严格下三角矩阵，$-U$ 为 A 的上三角部分构成的严格上三角矩阵. 当 B 分别取

(1) $\boldsymbol{B}_{\mathrm{J}} = \boldsymbol{D}^{-1}(\boldsymbol{L}+\boldsymbol{U})$,

(2) $\boldsymbol{B}_{\mathrm{G\text{-}S}} = (\boldsymbol{D}-\boldsymbol{L})^{-1}\boldsymbol{U}$,

(3) $\boldsymbol{B}_{\mathrm{SOR}} = (\boldsymbol{D}-\omega\boldsymbol{L})^{-1}((1-\omega)\boldsymbol{D}+\omega\boldsymbol{U}), 0<\omega<2$,

(4) $\boldsymbol{B}_{\mathrm{J}\omega} = (1-\omega)\boldsymbol{I}+\omega\boldsymbol{B}_{\mathrm{J}} = \boldsymbol{I}-\omega\boldsymbol{D}^{-1}\boldsymbol{A}, 0<\omega<1$

(13.1.17b)

时,可分别得到雅可比迭代,高斯-赛德尔迭代,SOR 迭代和雅可比-ω 迭代法(又称为阻尼雅可比迭代).迭代收敛的充分必要条件是 $\rho(\boldsymbol{B})<1$.迭代收敛的一种充分条件是 $\|\boldsymbol{B}\|<1$.称 $\rho(\boldsymbol{B})$ 为**渐近收敛因子**,称 $R(\boldsymbol{B})=-\ln\rho(\boldsymbol{B})$ 为**渐近收敛率**,称 $\|\boldsymbol{B}\|$ 为**收缩数**.

对 $\sigma=0$ 的一维离散模型(13.1.5)和(13.1.6),令 $\boldsymbol{A}=h^2\boldsymbol{L}_h$,$\boldsymbol{b}=h^2\boldsymbol{f}_h$,$h=1/N$,易求出 $\boldsymbol{A}$ 的第 k 个特征值和相应的第 k 个特征向量的第 j 个分量,它们分别为

$$\begin{cases}\lambda_k(\boldsymbol{A}) = 4\sin^2\dfrac{k\pi}{2N}, & 1\leqslant k\leqslant N-1,\\ w_{k,j} = \sin\dfrac{jk\pi}{N}, & 1\leqslant k\leqslant N-1,\end{cases} \tag{13.1.18}$$

其中 $w_{k,j}$ 也是矩阵 $\boldsymbol{A}$ 的傅里叶模.

雅可比迭代阵的特征值和特征向量的第 j 个分量分别为

$$\begin{cases}\lambda_k(\boldsymbol{B}_{\mathrm{J}}) = 1-2\sin^2\dfrac{k\pi}{2N}, & 1\leqslant k\leqslant N-1,\\ w_{k,j} = \sin\dfrac{jk\pi}{N}, & 1\leqslant k\leqslant N-1.\end{cases} \tag{13.1.19}$$

Jacobi-ω 迭代阵的特征值和特征向量的第 j 个分量分别为

$$\begin{cases}\lambda_k(\boldsymbol{B}_{\mathrm{J}})_\omega = 1-\dfrac{\omega\lambda(\boldsymbol{A})}{2} = 1-2\omega\sin^2\dfrac{k\pi}{2N}, & 1\leqslant k\leqslant N-1,\\ w_{k,j} = \sin\dfrac{jk\pi}{N}, & 1\leqslant k\leqslant N-1.\end{cases} \tag{13.1.20}$$

G-S 迭代阵的特征值和特征向量的第 j 个分量分别为

$$\begin{cases}\lambda_k(\boldsymbol{B}_{\text{G-S}}) = \cos^2\dfrac{k\pi}{N}, & 1 \leqslant k \leqslant N-1, \\ w_{k,j}^{\text{G-S}} = \cos^j\dfrac{k\pi}{N}\sin\dfrac{jk\pi}{N} = \lambda_k^{j/2}\sin\dfrac{jk\pi}{N}, & 1 \leqslant j \leqslant N, 1 \leqslant k \leqslant N-1.\end{cases} \tag{13.1.21}$$

下面先分析松弛对初始误差的光滑作用. 设迭代的初始误差向量用 $\boldsymbol{A}$ 的 $N-1$ 个线性无关的特征向量表示为

$$\boldsymbol{e}^{(0)} = \sum_{k=1}^{N-1} c_k \boldsymbol{w}_k, \quad c_k \in \mathbb{R},$$

用(13.1.17)式作 n 次迭代后,有

$$\boldsymbol{e}^{(n)} = \boldsymbol{B}^n \boldsymbol{e}^{(0)} = \sum_{k=1}^{N-1} c_k \lambda_k^n(\boldsymbol{B}) \boldsymbol{w}_k, \quad c_k \in \mathbb{R}, \tag{13.1.22}$$

这说明经 n 次迭代后,初始误差的第 k 个模减小到 $\lambda_k^n(\boldsymbol{B})\boldsymbol{w}_k$,把级数(13.1.22)的各项分成两部分,即高频分量和低频分量. 当 $n=8$ 时,示于图 13.2,其中 k 刚好给出 Ω_H 上"半波"的个数. 所谓低频

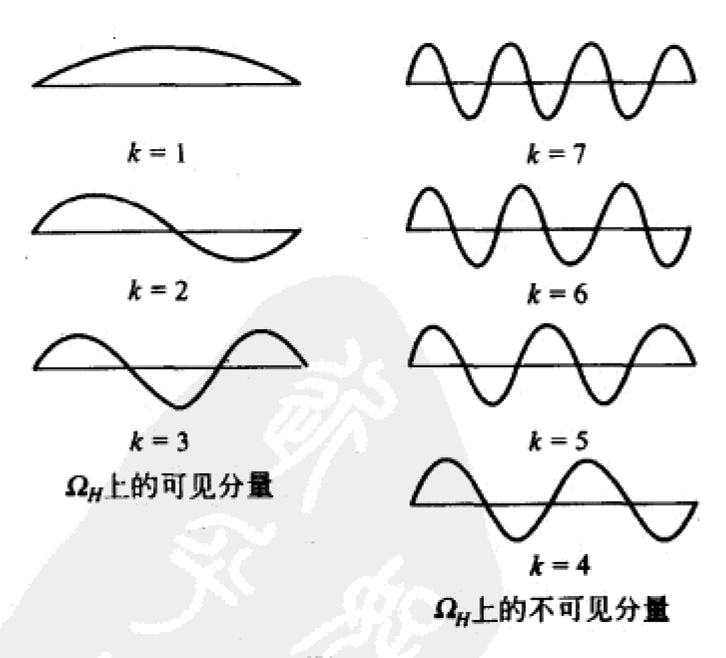


图 13.2

即在粗网 Ω_H 上能表现出来的那部分特征函数，而高频是在 Ω_H 上根本看不见的那部分特征函数. 具体定义为

低频：w_{kj}，$1\leqslant k<\dfrac{N}{2}$.

高频：w_{kj}，$\dfrac{N}{2}\leqslant k\leqslant N-1$.

为了度量松弛方法对误差的光滑程度，定义

$$\mu(\boldsymbol{B}) = \max_{N/2\leqslant k\leqslant N-1} |\lambda_k(\boldsymbol{B})|,$$

为松弛法的**光滑因子**(smoothing factor)，即松弛后消除误差的高频振荡分量，使误差变光滑，但并不是显著地减小误差. 图 13.3 为松弛法典型的光滑误差特性.

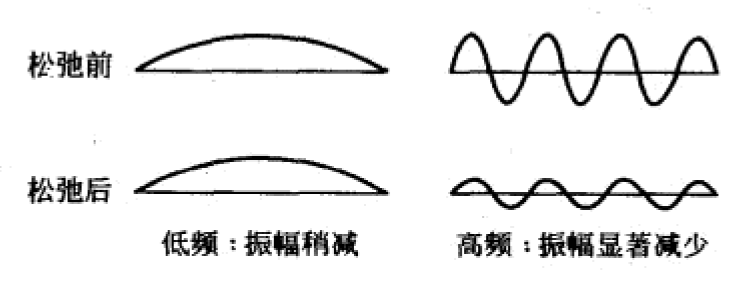


图 13.3

对阻尼雅可比方法，选 ω 使 $\max\{\lambda_{N/2}(\boldsymbol{B}_{J\omega}),\lambda_N(\boldsymbol{B}_{J\omega})\}$ 达到最小，即

$$\lambda_{N/2}(\boldsymbol{B}_{J\omega}) = -\lambda_N(\boldsymbol{B}_{J\omega}),$$

从而求出最佳参数 $\omega=2/3$. 图 13.4 表示 $\omega=1/3,1/2,2/3,1$ 时 $\lambda_k(\boldsymbol{B}_{J\omega})$ 随 k 变化的曲线，显然，$\omega=2/3$ 的阻尼雅可比法能最好地

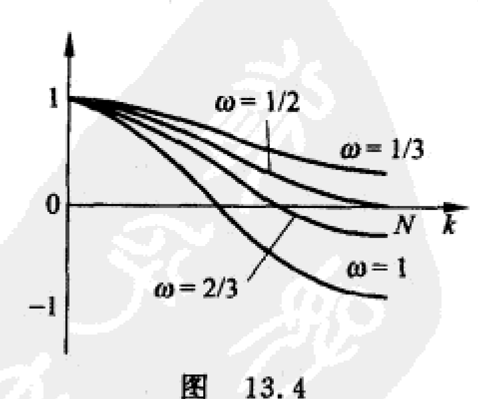


图 13.4

阻尼高频分量.这时

$$\lambda_k(\boldsymbol{B}_{J2/3}) = 1 - \frac{4}{3}\sin^2\frac{k\pi}{2N}, \quad 1 \leqslant k \leqslant N-1.$$

当 $N/2 \leqslant k \leqslant N-1$ 时,$|\lambda_k(\boldsymbol{B}_{J2/3})| \leqslant 1/3$,即有光滑因子 $\mu(\boldsymbol{B}_{J2/3}) = 1/3$,可见对误差的高频分量有很好的光滑作用.当 $0 < k < N/2$,$N=8$ 时,$1/2 < |\lambda_k(\boldsymbol{B}_{J2/3})| < 1$,因此,对误差的低频分量衰减不大.当 $\omega=1$ 时,雅可比-ω 迭代法就是雅可比迭代法,它对误差的高频和低频分量的衰减均很慢,仅对 $N/2$ 附近的波阻尼快,因此雅可比方法对误差的高频分量光滑效应差,在多重网格法中一般不采用.

再分析 G-S 迭代法,图 13.5 表示 $\lambda_k(\boldsymbol{B}_{G\text{-}S}) = \cos^2(k\pi/N)$ 随 k 变化的曲线,其特征向量

$$w_{kj}^{G\text{-}S} = \lambda_k^{j/2}\sin\frac{jk\pi}{N} = \lambda_k^{j/2}w_{kj},$$

则有

$$\boldsymbol{e}^{(0)} = \sum_{k=1}^{N-1} c_k \boldsymbol{w}_k^{G\text{-}S}, \quad \boldsymbol{e}^{(n)} = \boldsymbol{B}_{G\text{-}S}\boldsymbol{e}^{(0)} = \sum_{k=1}^{N-1} c_k \lambda_k^{n+j/2}(\boldsymbol{B}_{G\text{-}S})\boldsymbol{w}_k,$$

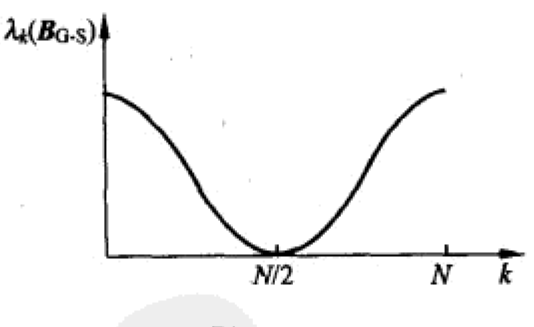


图 13.5

其中 $\boldsymbol{w}_k$ 为矩阵 $\boldsymbol{A}$ 的特征向量.要想解析地说明 G-S 迭代法对误差的高频分量的快衰减是困难的,从数值试验结果进行分析,对 $N=64$ 的一维模型,若以 $\boldsymbol{w}_k$ 作为初始猜测,则用 G-S 迭代将 $\boldsymbol{w}_k$ 减小到原来的 1/100 所需的迭代次数与波数 k 之间的关系曲线如图 13.6 所示.由此可见,G-S 迭代法对低频误差衰减很慢,对高频的振荡误差衰减很快.

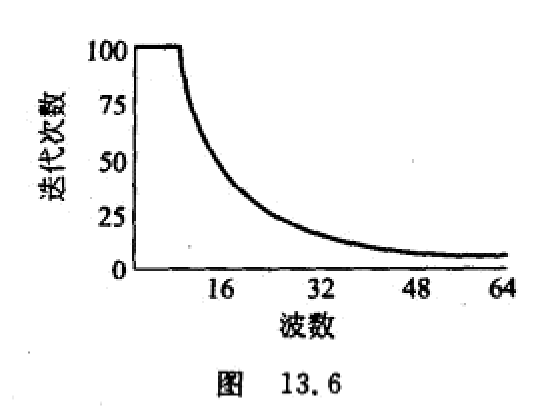


图 13.6

从以上分析可知，作几次阻尼雅可比迭代或 G-S 迭代均能有效地衰减误差的高频分量. 振荡模一旦被消去，留下光滑模，再继续迭代已无效. 弥补的办法之一是采用多重网格，特别是粗网修正. 因此，有转向粗网计算的必要性.

2. 粗网修正

先看谱半径

$$\rho(\boldsymbol{B}_{J\omega}) = \max_k \left| 1 - 2\omega \sin^2 \frac{k\pi}{2N} \right| \approx 1 - \frac{\omega\pi^2 h^2}{2} = 1 - O(h^2),$$

$$\rho(\boldsymbol{B}_J) = \max_k \left| 1 - 2\sin^2 \frac{k\pi}{2N} \right| \approx 1 - \frac{\pi^2 h^2}{2} = 1 - O(h^2),$$

$$\rho(\boldsymbol{B}_{G\text{-}S}) = \max_k \left| \cos^2 \frac{k\pi}{N} \right| = 1 - O(h^2),$$

同理有 $\rho(\boldsymbol{B}_{SOR}) = 1 - O(h)$. 从而在粗网上，由于网格变粗、$h$ 变大，迭代谱半径均变小，收敛率增加，因此，转向粗网有加速收敛的作用.

再看双网格法 CGC 中步骤②，从粗网转移到细网后的作用. 从图 13.7 可见，$\Omega_h(N=12)$ 上 $k=4$ 的波是低频波，限制转移到粗网 $\Omega_{2h}(N=6)$ 上是 $k=4$ 的高频波. 在粗网上松弛可将粗网上的高频模约化，即减小了细网上的低频模，比一直在细网上迭代消灭不了低频模更有效，使收敛速度加快，但 Ω_{2h} 上仍会留下它的低频模.

对应于双网格法的 CGC 中步骤③，在多重网格法中，这一步不是精确地求解，而是迭代求解亏损方程. 迭代同样只能光滑 Ω_{2h}

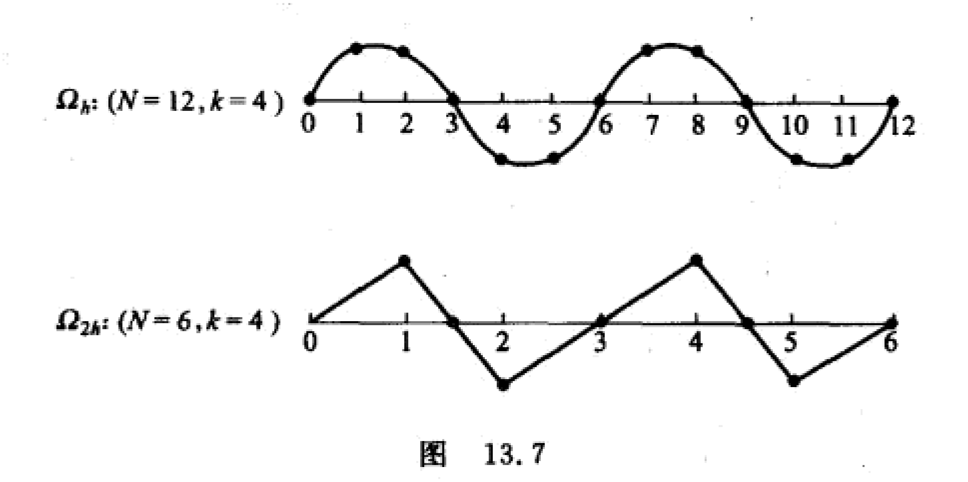


图 13.7

上的高频分量，同理只需迭代 1～2 次后便有转向更粗网格 Ω_{4h} 的必要. 重复这个过程，继续转向 Ω_{8h}，Ω_{16h} 等，直到最粗网格上的亏损量可精确地求解. 这是多重网格法与双网格法的不同之处.

在 CGC 步骤③中，为何求解亏损方程 $\boldsymbol{L}_H\boldsymbol{v}_H^{(n)}=\boldsymbol{d}_H^{(n)}$ 而不去求解粗网方程 $\boldsymbol{L}_H\boldsymbol{u}_H=\boldsymbol{f}_H$? 这是因为希望最终的亏损量是零向量，从而可用零向量作为迭代的初始量，这比起用 $\boldsymbol{u}_H^{(n)}=\boldsymbol{I}_h^H\boldsymbol{u}_h^{(n)}$ 作为初始向量去求解 $\boldsymbol{L}_H\boldsymbol{u}_H=\boldsymbol{f}_H$ 收敛速度快，也方便.

在 CGC 步骤④和⑤中，将粗网函数转移到细网及对细网解进行修正，这使细网解得到更好的近似值.

综上所述，多重网格法是把在细网上松弛消失误差中的高频振荡分量，使误差光滑化，并和在粗网上低频分量容易收敛及在粗网上修正为细网提供更好的初始值等相结合，形成一种特殊的收敛快的迭代过程.

例 13.1.3 用多重网格法计算一维模型问题 $\boldsymbol{Au}=\boldsymbol{0}(N=48)$ 的解. 取初始向量

$$\boldsymbol{u}_{h,j}^{(n)}=(\sin(12j\pi/N)+\sin(30j\pi/N))/2,$$

用 $\omega=2/3$ 的阻尼雅可比松弛，前松弛 $\nu_1=3$，后松弛 $\nu_2=3$.

设 $\boldsymbol{u}$ 为一维问题的精确解向量，$\boldsymbol{u}_h$ 为细网上的近似解向量，

$\boldsymbol{u}_{2h}$为粗网上的近似解向量,初始误差向量$\|\boldsymbol{e}\|_\infty=\|\boldsymbol{u}-\boldsymbol{u}_h\|_\infty=\|-\boldsymbol{u}_h\|=1.00$.

一个双网格迭代步的计算结果：

(1) 在细网上松弛 3 次后的误差为$\|\boldsymbol{e}_h\|_\infty=0.261$,这时误差的振荡分量基本消去.

(2) 用完全加权算子转移亏损量到粗网,在Ω_{2h}上对$\boldsymbol{A}_{2h}\boldsymbol{e}_{2h}=\boldsymbol{d}_{2h}$松弛 3 次后,有误差$\|\boldsymbol{e}_{2h}\|=0.0591$,为初始误差的 6%.

(3) 粗网修正后,再作 3 次细网松弛,有误差$\|\boldsymbol{e}_h\|_\infty=0.02$.

(4) 再把亏损量转移到粗网,又在Ω_{2h}上松弛 3 次后,有误差$\|\boldsymbol{e}_{2h}\|=0.00977$.

3. 双网格方法收敛性

定理 13.1.4 用双网格方法求解一维模型问题,选择$\omega=2/3$的阻尼雅可比松弛法,取$\nu_2=0$,$\boldsymbol{u}_h$为细网上的近似解向量,其初始误差向量为$\boldsymbol{e}_h$,经双网格法一个循环步以后的误差为$\bar{\boldsymbol{e}}_h$,则有误差估计

$$\|\bar{\boldsymbol{e}}_h\|_2^2\leqslant\left(\frac{1}{2}+\frac{1}{3^{\nu_1}}\right)\|\boldsymbol{e}_h\|_2^2.$$

当$\nu_1=2$时,

$$\|\bar{\boldsymbol{e}}_h\|_2\leqslant 0.782\|\boldsymbol{e}_h\|_2.$$

j次双网格步后的误差为

$$\|\boldsymbol{e}_h^{(j)}\|_2\leqslant 0.782^j\|\boldsymbol{e}_h^{(0)}\|_2.$$

因此,双网格方法以一个与h无关的收敛速率收敛于零.

双网格方法在实际计算中用得不多,但它是多重网格方法的理论基础,通过它阐述了多重网格法的基本原理.

13.2 线性多重网格法

多重网格法是在愈来愈粗的网格上递归地使用双网格方法.仅在最粗的网格上精确地求解差分方程.现在定义多重网格法所

用的符号和算子：

网格步长序列　$h_k(k=0,1,\cdots,M)$，通常 $h_k=h_{k-1}/2=2^{-k}h$.

网格序列　$\Omega_k(k=0,1,\cdots,M)$.

差分算子序列　$\boldsymbol{L}_k(k=0,1,\cdots,M)$，设 $\boldsymbol{L}_k^{-1}$ 存在.

松弛算子序列　$\boldsymbol{S}^{\nu}(\boldsymbol{u}_k,\boldsymbol{L}_k,\boldsymbol{f}_k)$.

限制算子序列　$\boldsymbol{I}_k^{k-1}(k=1,2,\cdots,M)$：$G(\Omega_k)\to G(\Omega_{k-1})$.

插值算子序列　$\boldsymbol{I}_{k-1}^{k}(k=1,2,\cdots,M)$：$G(\Omega_{k-1})\to G(\Omega_k)$.

其中 $G(\Omega_k)$为在 Ω_k 上网格函数的线性空间.

下面叙述用多重网格法迭代求解方程组

$$\boldsymbol{L}_M\boldsymbol{u}_M=\boldsymbol{f}_M \tag{13.2.1}$$

的计算过程. 设已知 $\boldsymbol{u}_M^{(n)}$，求 $\boldsymbol{u}_M^{(n+1)}$，即

$$\boldsymbol{u}_M^{(n+1)}\stackrel{\text{def}}{=}\mathrm{MR}_M(\boldsymbol{u}_M^{(n)},\boldsymbol{L}_M,\boldsymbol{f}_M), \tag{13.2.2}$$

其中 R＝V 为 **V 循环**，R＝W 为 **W 循环**. 仿照双网格方法，计算分为 3 个阶段.

(1) 前光滑：给定初值 $\boldsymbol{u}_M^{(n)}$，在 Ω_M 上作 $\nu_1(\geqslant 1)$次松弛得

$$\bar{\boldsymbol{u}}_M^{(n)}\stackrel{\text{def}}{=}\boldsymbol{S}^{\nu_1}(\boldsymbol{u}_M^{(n)},\boldsymbol{L}_M,\boldsymbol{f}_M).$$

(2) 粗网修正：

① 计算亏损量 $\boldsymbol{d}_M^{(n)}\stackrel{\text{def}}{=}\boldsymbol{f}_M-\boldsymbol{L}_M\bar{\boldsymbol{u}}_M^{(n)}$.

② 限制亏损量 $\boldsymbol{d}_{M-1}^{(n)}\stackrel{\text{def}}{=}\boldsymbol{I}_M^{M-1}\boldsymbol{d}_M^{(n)}$.

③ 在 Ω_{M-1}上近似求解亏损方程

$$\boldsymbol{L}_{M-1}\boldsymbol{v}_{M-1}^{(n)}=\boldsymbol{d}_{M-1}^{(n)}. \tag{13.2.3}$$

计算(13.2.3)式时，以零网格函数作为初始近似值，即$\boldsymbol{v}_{M-1}^{(n)}=\boldsymbol{0}$，作 $M-1$ 网格法的 r 型迭代. $M-1$ 网格法用 $\Omega_{M-1},\Omega_{M-2},\cdots,\Omega_0$，即

$$\boldsymbol{v}_{M-1}^{(n)}\stackrel{\text{def}}{=}\mathrm{MR}_{M-1}(\boldsymbol{v}_{M-1}^{(n)},\boldsymbol{L}_{M-1},\boldsymbol{d}_{M-1}).$$

在最粗的网格上精确求解

$$\boldsymbol{L}_0\boldsymbol{v}_0=\boldsymbol{d}_0.$$

④ 插值$\boldsymbol{v}_M^{(n)}\stackrel{\text{def}}{=}\boldsymbol{I}_{M-1}^{M}\boldsymbol{v}_{M-1}^{(n)}$.

⑤ 修正 Ω_M 上的节点函数值 $\hat{\boldsymbol{u}}_M^{(n)}=\boldsymbol{u}_M^{(n)}+\boldsymbol{v}_M^{(n)}$.

(3) 后光滑：在 Ω_M 上作 $\nu_2(\geqslant 1)$ 次迭代得

$$\boldsymbol{u}_M^{(n+1)} \overset{\text{def}}{=} \boldsymbol{S}^{\nu_2}(\hat{\boldsymbol{u}}_M^{(n)}, \boldsymbol{L}_M, \boldsymbol{f}_M).$$

上面粗网修正中的 r 也是循环参数，一般 $r=1,2$ 或 3，当 $r>3$ 时多重网格法就不太有效了. 我们称 $r=1$ 为 **V 循环**，$r=2$ 为 **W 循环**，$r=3$ 很少见.

用图 13.8 的符号分别表示 $M=1,2,3$ 时，$r=1,2,3$ 的多重网格法的一个循环(或一个迭代步). 图 13.9 的流程图表示一个多重网格迭代步的计算过程，其中引入了一个开关参数 $C(k)$，$0\leqslant C(k)\leqslant r$，以控制转向粗网和返回细网.

多重网格法的迭代公式是

$$\boldsymbol{u}_M^{(n+1)} = \boldsymbol{M}_M \boldsymbol{u}_M^{(n)} + \boldsymbol{N}_M(\boldsymbol{f}_M),$$

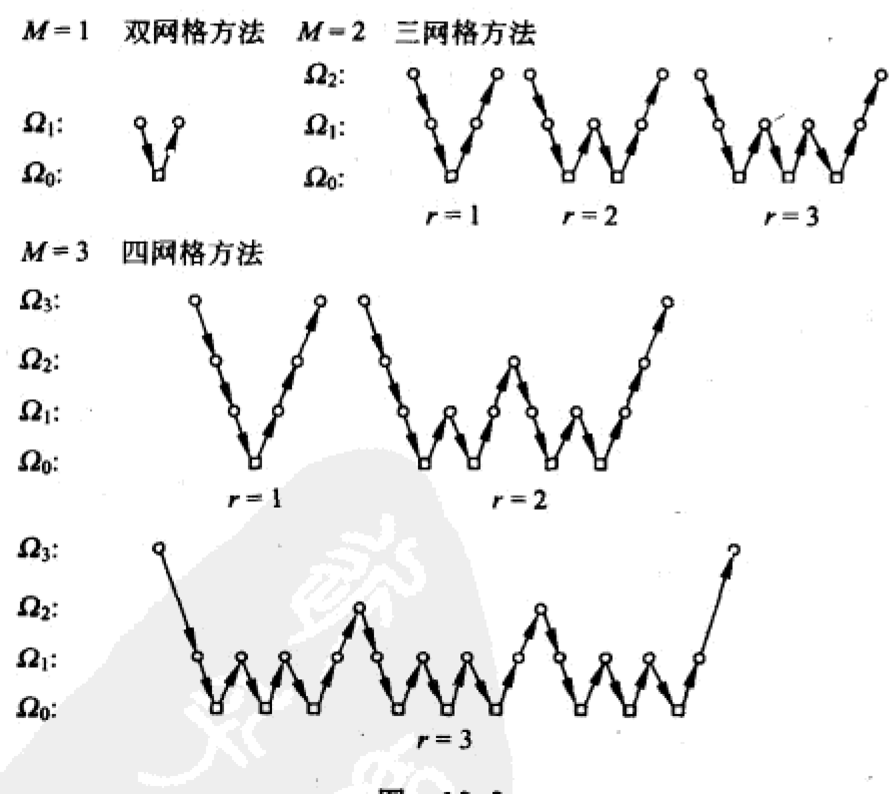


图 13.8

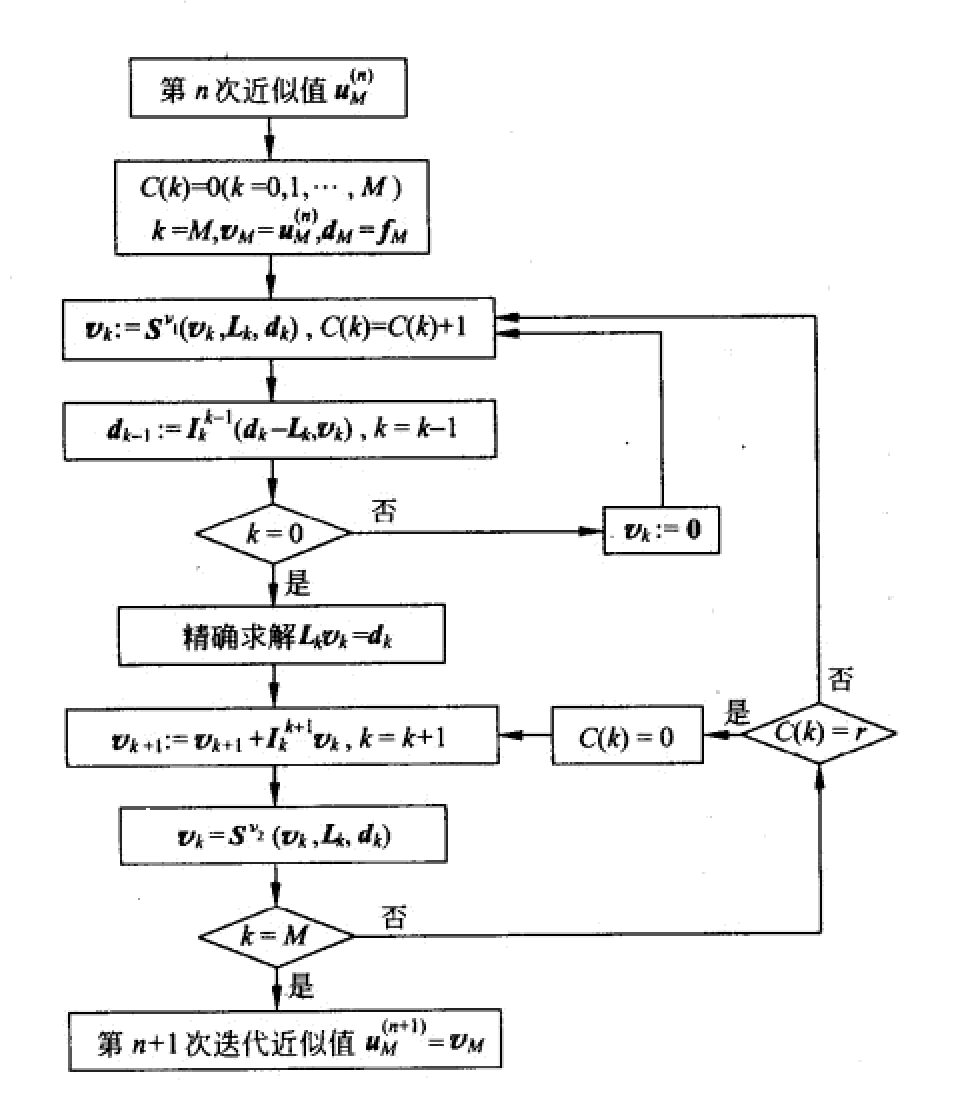


图 13.9

其中

$$M_1 \overset{\text{def}}{=} S_1^{\nu_2}(I_1 - I_0^1 L_0^{-1} I_1^0 L_1) S_1^{\nu_1}$$

$$M_k \overset{\text{def}}{=} S_k^{\nu_2}(I_k - I_{k-1}^k (I_{k-1} - A_{k-1}^r) L_{k-1}^{-1} I_k^{k-1} L_k) S_k^{\nu_1},$$

$$k = 2,3,\cdots,M. \tag{13.2.4}$$

而 Ω_{M-1} 和 Ω_M 的双网格算子为

$$M_M^{M-1} \overset{\text{def}}{=} S_M^{\nu_2}(I_M - I_{M-1}^M L_{M-1}^{-1} I_M^{M-1} L_M) S_M^{\nu_1}. \tag{13.2.5}$$

式(13.2.4)与式(13.2.5)的差别在于用 $(I_{M-1} - A_{M-1}^r) L_{M-1}^{-1}$ 代替

式(13.2.5)中的 L_{M-1}^{-1},这说明用 $\Omega_{M-1},\Omega_{M-2},\cdots,\Omega_0$ 的 $M-1$ 网格 r 型多重网格迭代去近似求解

$$L_{M-1}\,v_{M-1}^{(n)}=d_{M-1}^{(n)}.$$

以上是线性多重网格法的计算步骤.关于多重网格法的收敛性,一般来说,若双网格方法收敛很好,则相应的 $r=2$ 的多重网格法亦收敛得很好.

13.3 完整的多重网格法

多重网格法与完整网格技巧相结合,或者说与套迭代技术相结合,称为**完整的多重网格法**(full multigrid method,简称 FMG 方法),FMG 方法效果更明显.所谓套迭代就是粗网为细网提供良好的初值.这是古典迭代方法(例如 SOR 方法)不可能做到的.

FMG 方法的计算步骤和流程图分别示于图 13.10 和图 13.11 中.在图 13.11 中,INT 表示插值,MGI 表示在 $\Omega_k(k=0,1,\cdots,M)$上用多重网格 r 型迭代求解方程 $L_k u_k=f_k(k=0,1,\cdots,M)$.当 $r=1$ 时为 V 循环,称为 **FMV 方法**,$r=2$ 时为 W 循环,称为 **FMW 方法**.在图 13.11 中,当 $M=6,r=1$ 时为图 13.10 所示的 FMV 过程.

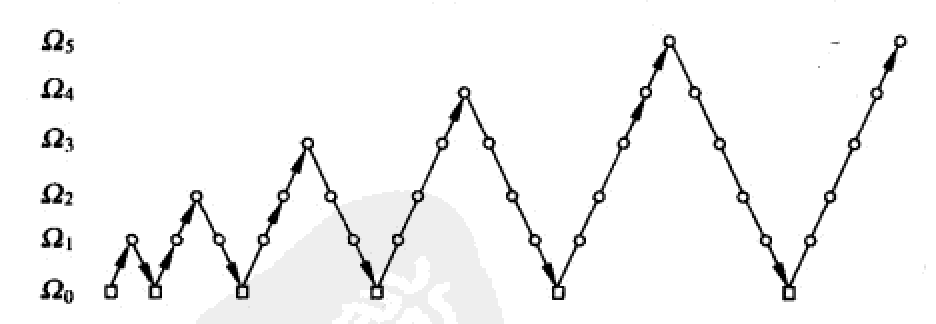


图 13.10 FMG 方法计算步骤

由式(13.2.2),一个多重网格法的 V 循环步为

$$u_M^{(n+1)}:=\mathrm{MV}_M(u_M^{(n)},L_M,f_M),$$

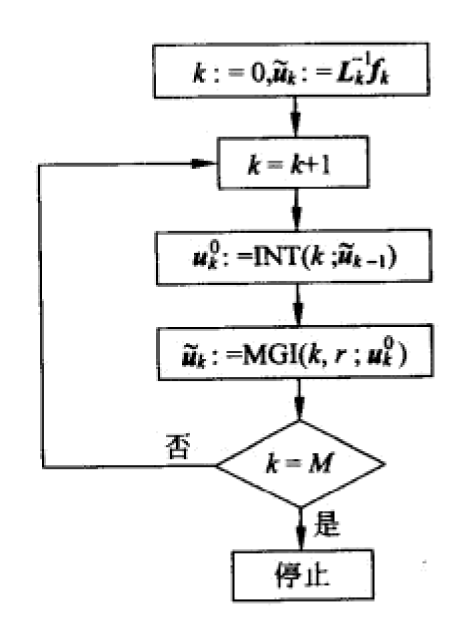


图 13.11　FMG 方法流程图

一个 FMV 的循环步定义为

$$u_M^{(n+1)} := \mathrm{FMV}_M(u_M^{(n)}, L_M, f_M)$$

其计算步骤如下：

将 $f_M, f_{M-1}, f_{M-2}, \cdots$ 及 $u_M, u_{M-1}, u_{M-2}, \cdots$ 初始化为零. 在最粗网格上松弛或直接求解方程 $A_0 u_0 = f_0$，

$$\vdots$$

$$u_{M-2} = u_{M-2} + I_{M-3}^{M-2} u_{M-3},$$

$$u_{M-2} = \mathrm{MV}_{M-2}(u_{M-2}, L_{M-2}, f_{M-2}),$$

$$u_{M-1} = u_{M-1} + I_{M-2}^{M-1} u_{M-2},$$

$$u_{M-1} = \mathrm{MV}_{M-1}(u_{M-1}, L_{M-1}, f_{M-1}),$$

$$u_M = u_M + I_{M-1}^{M} u_{M-1},$$

$$u_M = \mathrm{MV}_M(u_M, L_M, f_M).$$

FMV 循环的递归如下：

(1) 如果 Ω_M 为最粗网格，则转到步骤(3)，否则

$$f_{M-1} = I_M^{M-1}(f_M - A_M u_M),$$

$$u_{M-1}=\mathbf{0},$$

$$u_{M-1}=\mathrm{FMV}_{M-1}(u_{M-1},L_{M-1},f_{M-1}).$$

(2) 修正 $u_M=u_M+I_{M-1}^{M}u_{M-1}$.

(3) $u_M=\mathrm{MV}_M(u_M,L_M,f_M)$.

13.4 二维多重网格诸元素

13.4.1 差分格式

对二维模型问题(例 13.1.2),熟知的五点差分格式是式(13.1.7).当 $h_x=h_y=h$ 时,对每点(x_i,y_j)上的函数值 u_{ij} 与周围邻点的关系可表示为

$$\begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix}u_k(x,y)=h_k^2 f_k(x,y),\quad \forall (x,y)\in\Omega_k. \tag{13.4.1}$$

对九点差分格式,可用九点星表示:

$$\frac{1}{6}\begin{bmatrix} -1 & -4 & -1 \\ -4 & 20 & -4 \\ -1 & -4 & -1 \end{bmatrix}u_k(x,y)=\frac{h_k^2}{12}\begin{bmatrix} & 1 & \\ 1 & 8 & 1 \\ & 1 & \end{bmatrix}f_k(x,y). \tag{13.4.2}$$

13.4.2 光滑迭代

式(13.1.17)中所列的几种迭代方法对二维泊松方程边值问题例 13.1.2 均有效,对奇异摄动等特殊问题需要适当选取光滑过程.下面讨论几种经典的迭代.

1. G-S 迭代

多维 G-S 迭代与未知量的编号次序有关.对于二维情况,通

过 Ω_k 上网点即 $\boldsymbol{x}^i=(x^i,y^i)$ 的排列次序确定未知量 $u_{k,i}:=u_k(\boldsymbol{x}^i)$ 的顺序.

字典顺序：x 方向上的点由左到右，y 方向上的线由下往上逐点排列，见图 13.12.

旋转字典顺序：y 方向上的点由下往上，x 方向上的线由左往右逐点排列，见图 13.13.

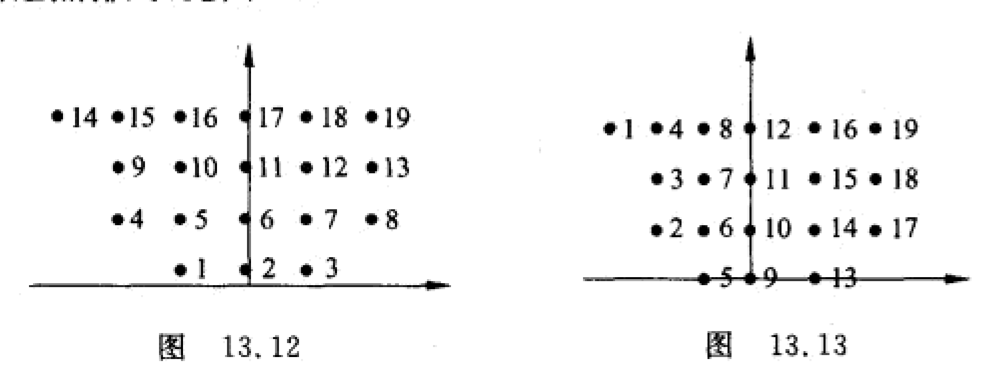


图 13.12 图 13.13

红-黑顺序：设 $\Omega_k=\{(jh_k,lh_k)\mid j,l\in\mathbb{Z}\}$，把 Ω_k 中的点分成两类，称 $j+l$ 为偶数的点为红点，$\Omega_k^r=\{(jh_k,lh_k),j+l$ 为偶数$\}$，用■表示；$\Omega_k^b=\Omega_k/\Omega_k^r$ 的点为黑点，用●表示，见图 13.14.

斑马顺序：把 j 为偶数的线称为红线，用●表示；j 为奇数的线称为黑线，用■表示，见图 13.15.

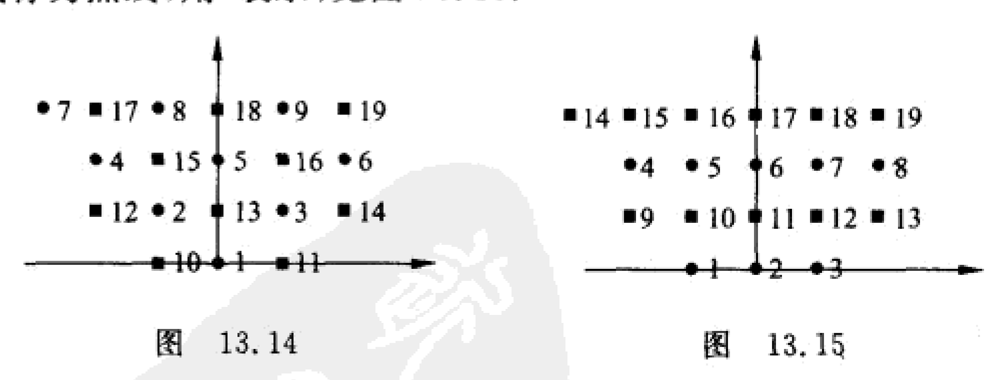


图 13.14 图 13.15

四色顺序：把 Ω_k 分成 4 个集，

$\Omega_k^1=\{(jh_k,lh_k)\mid j,l$ 为偶数$\}$，用●表示.

$\Omega_k^2=\{(jh_k,lh_k)\mid j,l$ 为奇数$\}$，用■表示.

$\Omega_k^3=\{(jh_k,lh_k)\mid j$ 为偶数，l 为奇数$\}$，用⊗表示.

$\Omega_k^4=\{(jh_k,lh_k)\mid j$ 为奇数，l 为偶数$\}$，用×表示.

按字典顺序依次排列 $\Omega_k^1,\Omega_k^2,\Omega_k^3$ 和 Ω_k^4 中的网点，见图 13.16.

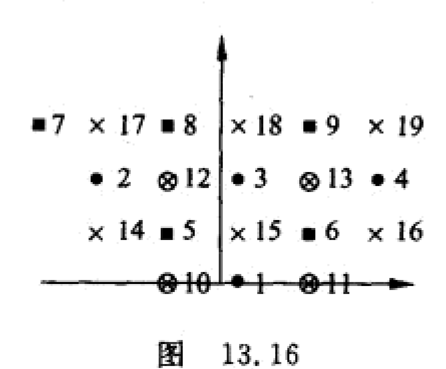


图 13.16

(1) 点 G-S 迭代

字典顺序和旋转字典顺序适合于逐点 G-S 迭代，迭代公式为

$$\frac{-u_{i-1,j}^{(n+1)}+2u_{i,j}^{(n+1)}-u_{i+1,j}^{(n)}}{h_x^2}+\frac{-u_{i,j-1}^{(n+1)}+2u_{i,j}^{(n+1)}-u_{i,j+1}^{(n)}}{h_y^2}=f_{i,j},\tag{13.4.3}$$

其中 $u_{i,j}^{(n+1)}$ 为待求值，$u_{i-1,j}^{(n+1)},u_{i,j-1}^{(n+1)}$ 为已求出的新值，$u_{i+1,j}^{(n)},u_{i,j+1}^{(n)}$ 为上一次迭代之值.

红黑顺序适合于五点差分格式，先依次算红点，再依次算黑点，其迭代公式为

$$\frac{-u_{i-1,j}^{(n)}+2u_{i,j}^{(n+1)}-u_{i+1,j}^{(n)}}{h_x^2}+\frac{-u_{i,j-1}^{(n)}+2u_{i,j}^{(n+1)}-u_{i,j+1}^{(n)}}{h_y^2}=f_{i,j},\tag{13.4.4}$$

其中红点上的 $u_{i,j}^{(n+1)}$ 为待求量，其相邻的黑点均为前一次的迭代值，黑点上的量为已知量. 下一轮计算时，黑点上的量为待求量，其相邻的红点上均为前一次的迭代值，这时，红点上的量为已知量.

四色顺序适合于九点差分格式. 先依次算 Ω_k^1 中的点，接着依次算 Ω_k^2 中的点，再依次算 Ω_k^3 中的点，最后依次算 Ω_k^4 中的点，这样构成一次迭代.

Ω_k^1 中点的 G-S 迭代公式为

$$\frac{1}{6h^2}\Big[20u_{i,j}^{(n+1)}-(u_{i-1,j-1}^{(n)}+u_{i-1,j+1}^{(n)}+u_{i+1,j-1}^{(n)}+u_{i+1,j+1}^{(n)})-$$

$$4(u_{i-1,j}^{(n)}+u_{i+1,j}^{(n)}+u_{i,j-1}^{(n)}+u_{i,j+1}^{(n)})\Big]$$

$$=\frac{1}{12}(8f_{i,j}^{(n)}+f_{i-1,j}^{(n)}+f_{i+1,j}^{(n)}+f_{i,j-1}^{(n)}+f_{i,j+1}^{(n)}).$$

Ω_k^2 中点的 G-S 迭代公式为

$$\frac{1}{6h^2}\Big[20u_{i,j}^{(n+1)}-(u_{i-1,j-1}^{(n+1)}+u_{i-1,j+1}^{(n+1)}+u_{i+1,j-1}^{(n+1)}+u_{i+1,j+1}^{(n+1)})-$$

$$4(u_{i-1,j}^{(n)}+u_{i+1,j}^{(n)}+u_{i,j-1}^{(n)}+u_{i,j+1}^{(n)})\Big]$$

$$=\frac{1}{12}(8f_{i,j}^{(n)}+f_{i-1,j}^{(n)}+f_{i+1,j}^{(n)}+f_{i,j-1}^{(n)}+f_{i,j+1}^{(n)}).$$

Ω_k^3 中点的 G-S 迭代公式为

$$\frac{1}{6h^2}\Big[20u_{i,j}^{(n+1)}-(u_{i-1,j-1}^{(n)}+u_{i-1,j+1}^{(n)}+u_{i+1,j-1}^{(n)}+u_{i+1,j+1}^{(n)})-$$

$$4(u_{i-1,j}^{(n+1)}+u_{i+1,j}^{(n+1)}+u_{i,j-1}^{(n+1)}+u_{i,j+1}^{(n+1)})\Big]$$

$$=\frac{1}{12}(8f_{i,j}^{(n)}+f_{i-1,j}^{(n+1)}+f_{i+1,j}^{(n+1)}+f_{i,j-1}^{(n+1)}+f_{i,j+1}^{(n+1)}).$$

Ω_k^4 中点的 G-S 迭代公式为

$$\frac{1}{6h^2}\Big[20u_{i,j}^{(n+1)}-(u_{i-1,j-1}^{(n+1)}+u_{i-1,j+1}^{(n+1)}+u_{i+1,j-1}^{(n+1)}+u_{i+1,j+1}^{(n+1)})-$$

$$4(u_{i-1,j}^{(n+1)}+u_{i+1,j}^{(n+1)}+u_{i,j-1}^{(n+1)}+u_{i,j+1}^{(n+1)})\Big]$$

$$=\frac{1}{12}(8f_{i,j}^{(n)}+f_{i-1,j}^{(n+1)}+f_{i+1,j}^{(n+1)}+f_{i,j-1}^{(n+1)}+f_{i,j+1}^{(n+1)}).$$

在以上 4 个公式中仅 $u_{i,j}^{(n+1)}$ 为未知量，其余均为已知量，仍属点 G-S 公式. 每个公式中待求点与周围邻点的关系见图 13.17.

■ × ■　　● ⊗ ●　　× ■ ×　　⊗ ● ⊗
⊗ ● ⊗　　× ■ ×　　● ⊗ ●　　■ × ■
■ × ■　　● ⊗ ●　　× ■ ×　　⊗ ● ⊗

Ω_k^1的邻点　　Ω_k^2的邻点　　Ω_k^3的邻点　　Ω_k^4的邻点

图 13.17

(2) 块 G-S 迭代

如果将图 13.12 中的字典顺序网点进行分组，在 Ω_k 上，y_j 为常量的每条线上的网点为一组，共分成四组，分别为 $I_k^1=\{1,2,3\}$，$I_k^2=\{4,5,6,7,8\}$，$I_k^3=\{9,10,11,12,13\}$，$I_k^4=\{14,15,16,17,18,19\}$，对五点差分格式，类似于式(13.1.8)，可得矩阵方程组

$$\begin{pmatrix} \boldsymbol{A}_1 & -\boldsymbol{I} & & \\ -\boldsymbol{I} & \boldsymbol{A}_2 & -\boldsymbol{I} & \\ & -\boldsymbol{I} & \boldsymbol{A}_3 & -\boldsymbol{I} \\ & & -\boldsymbol{I} & \boldsymbol{A}_4 \end{pmatrix} \begin{pmatrix} \boldsymbol{u}_k^1 \\ \boldsymbol{u}_k^2 \\ \boldsymbol{u}_k^3 \\ \boldsymbol{u}_k^4 \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_k^1 \\ \boldsymbol{f}_k^2 \\ \boldsymbol{f}_k^3 \\ \boldsymbol{f}_k^4 \end{pmatrix},$$

其中 $\boldsymbol{u}_k^1, \boldsymbol{f}_k^1 \in \mathbb{R}^3$，$\boldsymbol{A}_1 \in \mathbb{R}^{3\times 3}$，$\boldsymbol{u}_k^2, \boldsymbol{f}_k^2, \boldsymbol{u}_k^3, \boldsymbol{f}_k^3 \in \mathbb{R}^5$，$\boldsymbol{A}_2, \boldsymbol{A}_3 \in \mathbb{R}^{5\times 5}$，$\boldsymbol{u}_k^4, \boldsymbol{f}_k^4 \in \mathbb{R}^6$，$\boldsymbol{A}_4 \in \mathbb{R}^{6\times 6}$，$\boldsymbol{A}_i(i=1,2,3,4)$ 均为三对角矩阵，从而形成一次求解一条线上未知量的块迭代公式：

$$\boldsymbol{A}_i \boldsymbol{u}_k^{i(n+1)} = a_i \boldsymbol{u}_k^{(i-1)(n+1)} + b_i \boldsymbol{u}_k^{(i+1)(n)} + \boldsymbol{f}_k^i \quad (i = 1,2,3,4),$$

其中 $a_0=b_4=0$，$a_1=a_2=a_3=1$，$b_1=b_2=b_3=1$. 上式右端均为已知向量，可直接用追赶法求解此方程组．称以上的块迭代法为**字典 *x*-线 G-S 迭代法**.

若用旋转字典顺序，在 Ω_k 上，x_j 为常量的每条线上的网点为一组，则类似地可构成**字典 *y*-线 G-S 迭代法**.

按斑马顺序，如图 13.15 构造相应的 G-S 迭代称为 ***x*-斑马线 G-S 迭代法**. 同理可构造 ***y*-斑马线 G-S 迭代法**.

在通常情况下，线 G-S 迭代法的光滑性质稍比点 G-S 迭代法好.

2. 阻尼雅可比方法

对于二维模型例 13.1.2，其离散方程为式(13.1.7)和式(13.1.8). 令 $\boldsymbol{A}=h^2\boldsymbol{L}_h$，当 $h_x=h_y=h=1/N$ 时，$\boldsymbol{A}$ 的特征值和特征向量分别为

$$\lambda_{\boldsymbol{k}}(\boldsymbol{A}) = 4\left(1-\frac{1}{2}(\cos(k_1\pi h)+\cos(k_2\pi h))\right),$$

$$1 \leqslant k_1, k_2 \leqslant N-1, \quad (13.4.5)$$

$$w_{k,j}(A) = 2\sin(k_1\pi x)\sin(k_2\pi y), \quad 1 \leqslant k_1, k_2 \leqslant N-1. \tag{13.4.6}$$

令 $x=(x,y)\in\Omega_h$.

阻尼雅可比松弛公式是

$$\bar{u}_h^{(n)} = u_h^{(n)} + \omega(z_h(x) - u_h^{(n)}), \tag{13.4.7}$$

$$\frac{4}{h^2}z_h(x) + L_h u_h^{(n)} - \frac{4}{h^2}u_h^{(n)} = f_h(x), \quad x\in\Omega_h, \tag{13.4.8}$$

其迭代矩阵为

$$B_{J\omega} = I_h - \frac{\omega}{4}A, \tag{13.4.9}$$

易知其特征值和特征向量分别为

$$\lambda_k(B_{J\omega}) = 1 - \frac{\omega}{2}\left(2 - \cos\frac{k_1\pi}{N} - \cos\frac{k_2\pi}{N}\right), \quad 1 \leqslant k_1, k_2 \leqslant N-1, \tag{13.4.10}$$

$$w_k(B_{J\omega}) = 2\sin(k_1\pi x)\sin(k_2\pi y), \quad 1 \leqslant k_1, k_2 \leqslant N-1; \tag{13.4.11}$$

其中 $k=(k_1,k_2)$. 若迭代初始误差按 $B_{J\omega}$ 的特征向量 $w_k(B_{J\omega})$ 展开，则有

$$e^{(0)} = \sum_{|k|\leqslant N-1} d_k w_k(B_{J\omega}),$$

$$|k| = \max(k_1, k_2),$$

其中 d_k 为常数，经 n 次迭代后

$$e^{(n)} = \sum_{|k|\leqslant N-1} d_k \lambda_k^n(B_{J\omega}) w_k(B_{J\omega}).$$

把级数分成高频和低频两部分

低频：$w_k, |k| < N/2$,

高频：$w_k, N/2 \leqslant |k| \leqslant N-1$,

选 ω 使 $\max\{\lambda_{N/2}(B_{J\omega}), \lambda_N(B_{J\omega})\}$ 达到最小，即

$$\lambda_{N/2}(B_{J\omega}) = -\lambda_N(B_{J\omega}),$$

求出使阻尼雅可比迭代法收敛最快的最佳松弛因子 $\omega_{opt}=2/3$. 同一维一样，$\omega_{opt}=2/3$ 的雅可比迭代法能再次很好地阻尼高频分量，这时

$$\lambda_k(\boldsymbol{B}_{J\omega}) = 1 - \frac{4}{3}\sin^2\left(\frac{k_1\pi}{2N}\sin^2\frac{k_2\pi}{2N}\right).$$

当 $N/2 \leqslant |\boldsymbol{k}| \leqslant N-1$ 时，$|\lambda_k(\boldsymbol{B}_{J2/3})| \leqslant 1/3$，即阻尼雅可比迭代法的光滑因子 $\mu(\boldsymbol{B}_{J2/3})=1/3$，故它对高频分量有很好的光滑作用.

阻尼雅可比迭代法与未知量的排列顺序无关.

对于块阻尼雅可比法，仅有阻尼 x-线雅可比迭代法和阻尼 y-线雅可比迭代法，它们均与块的排列顺序无关.

13.4.3 网格粗化

常用的**网格粗化**方式有：

标准粗化：$h_{k-1}=2h_k$（见图 13.18）. 只要求 $h_{k-1}^x/h_k^x=h_{k-1}^y/h_k^y=2$，可以假定 x 和 y 方向上的网格步长 h_k^x 和 h_k^y 是不同的.

半粗化：x 粗化，由 $h_{k-1}^y=h_k^y$ 和 $h_{k-1}^x/h_k^x=2$ 定义粗网格，或 y 粗化，由 $h_{k-1}^x=h_k^x$ 和 $h_{k-1}^y/h_k^y=2$ 定义粗网格（见图 13.19）.

红黑粗化：仅考虑方网格，即 $h_k^x=h_k^y$，又称为$\sqrt{2}$的均匀网格粗化（见图 13.20）.

图 13.18　　图 13.19　　图 13.20

13.4.4 延拓或插值

1. 分片线性插值作为延拓

对二阶或四阶微分方程边值问题，一般用分片线性插值就足够了. 这里只介绍九点延拓和七点延拓.

(1) 九点延拓

用如下的九点星模式符号描述：

$$I_{k-1}^{k} \stackrel{\text{def}}{=} \frac{1}{4}\left]\begin{matrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{matrix}\right[,$$

它表示粗网上的函数值按此权系数分配到邻近的细网点上去. 取一个标准单元$(0,2h)\times(0,2h)$为例. 如果粗细网点重合，则在该点细网函数值就等于粗网点函数值，即

$$\begin{cases} u_k(0,0)=u_{k-1}(0,0), \quad u_k(0,2h)=u_{k-1}(0,2h), \\ u_k(2h,0)=u_{k-1}(2h,0), \quad u_k(2h,2h)=u_{k-1}(2h,2h). \end{cases}$$

若细网点位于两个粗网点之间，则用这两个粗网点上的函数值作线性插值，就得到细网点上的函数值，

$$\begin{cases} u_k(0,h)=(u_{k-1}(0,0)+u_{k-1}(0,2h))/2, \\ u_k(h,0)=(u_{k-1}(0,0)+u_{k-1}(2h,0))/2, \\ u_k(2h,h)=(u_{k-1}(2h,0)+u_{k-1}(2h,2h))/2, \\ u_k(h,2h)=(u_{k-1}(0,2h)+u_{k-1}(2h,2h))/2. \end{cases}$$

有两种插值方法求位于四个粗网点中心的细网点函数值，即

$$\begin{aligned} u_k(h,h)=(&u_{k-1}(0,0)+u_{k-1}(2h,0)+ \\ &u_{k-1}(0,2h)+u_{k-1}(2h,2h))/4 \end{aligned}$$

或 $$u_k(h,h)=(u_{k-1}(0,0)+u_{k-1}(2h,2h))/2.$$

对于其他网格单元$(2ih_k,2ih_k+2h_k)\times(2jh_k,2jh_k+2h_k)$，可得到类似的双线性插值公式.

九点延拓对应于双线性有限元. 对于 $0\leqslant x, y\leqslant 2h$，$u_k=I_{k-1}^{k}u_{k-1}$的$u_k(x,y)$与$u_{k-1}$在$(0,0)$，$(2h,0)$，$(0,2h)$，$(2h,2h)$点上函数值的双线性插值相同.

(2) 七点延拓

用如下的七点星模式符号描述：

$$I_{k-1}^{k} \stackrel{\text{def}}{=} \left]\begin{matrix} 0 & 1/2 & 1/2 \\ 1/2 & 1 & 1/2 \\ 1/2 & 1/2 & 0 \end{matrix}\right[,$$

七点延拓对应于三角形上的线性有限元. 对于 $0 \leqslant x, y \leqslant 2h$, $u_k = I_{k-1}^{k} u_{k-1}$ 的 $u_k(x,y)$ 与 u_{k-1} 在 $(0,0)$, $(0,2h)$, $(2h,2h)$ 点上函数值的线性插值相同.

分片线性插值推广到三维或更高维是显然的.

2. 高阶插值

对于四阶微分方程边值问题，通常用分片二次延拓是足够的，而用三次插值更好些，因为它并不比二次插值公式复杂.

若已知粗网上的插值节点 $(x-3h_k, y)$, $(x-h_k, y)$, $(x+h_k, y)$, $(x+3h_k, y)$，则插值点 $(x,y)=(ih_k, jh_k)$（i 为奇数，j 为偶数）的三次拉格朗日插值 $(I_{k-1}^{k} u_{k-1})(x,y)$ 为

$$u_k(x,y) = -\frac{1}{16}[u_{k-1}(x-3h_k, y) + (x+3h_k, y)] + \frac{9}{16}[u_{k-1}(x-h_k, y) + (u_{k-1}(x+h_k, y))],$$
$$x \in [x-3h_k, x+3h_k]. \tag{13.4.12}$$

当点 (x,y) 中的 j 为奇数时，我们能重复 y 方向的插值过程. 由插值公式的余项可知，三次拉格朗日插值对于任一双三次函数 $\sum_{i=0}^{3}\sum_{j=0}^{3} a_{ij} x^i y^j$ 是精确的.

如果某个插值点有一个或两个邻点不在域内，则不能用三次插值公式，如在边界附近，取边界点 $(x-sh_k, y)$ 为插值节点，再取两个内部粗网点 $(x+h_k, y)$, $(x+3h_k, y)$ 为插值节点并且边界条件是齐次狄利克雷条件 $u(x-sh_k, y)=0$，那么用如下的二次拉格朗日插值公式去求细网点上的函数值 $u_k(x,y)$ 和 $u_k(x+2h_k, y)$ 更方便：

$$u_k(x,y) = \frac{s}{1+s} \cdot \frac{3}{2} u_{k-1}(x+h_k, y) - \frac{s}{3+s} \cdot \frac{1}{2} u_{k-1}(x+3h_k, y),$$

$$u_k(x+2h_k, y) = \frac{2+s}{1+s} \cdot \frac{1}{2} u_{k-1}(x+h_k, y) + \frac{2+s}{3+s} \cdot \frac{1}{2} u_{k-1}(x+3h_k, y).$$

在区域内部的单边二次插值公式为

$$u_k(x,y) = -\frac{1}{8} u_{k-1}(x-3h_k, y) + \frac{3}{4} u_{k-1}(x-h_k, y) + \frac{3}{8} u_k(x+h_k, y).$$

13.4.5 限制

1. 九点限制

用如下的九点星模式符号描述：

$$I_k^{k-1} \stackrel{\text{def}}{=} \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}.$$

九点限制即完全加权算子，若有粗网点$(2h,2h)$，按完全加权算子运算，该粗网点上的函数值与周围细网点上函数值的关系如下：

$$u_{k-1}(2h,2h) = \frac{1}{16}\Big[4u_k(2h,2h) + 2(u_k(2h,h) + u_k(h,2h) + u_k(3h,2h) + u_k(2h,3h)) + u_k(h,h) + u_k(3h,h) + u_k(h,3h) + u_k(3h,3h)\Big].$$

2. 七点限制

用如下的七点星模式符号描述：

$$I_k^{k-1} \stackrel{\text{def}}{=} \frac{1}{4} \begin{bmatrix} 0 & 1/2 & 1/2 \\ 1/2 & 1 & 1/2 \\ 1/2 & 1/2 & 0 \end{bmatrix}.$$

显然，按七点加权运算，粗网点$(2h,2h)$上的函数值与周围细网点上函数值的关系如下：

$$u_{k-1}(2h,2h)=\frac{1}{4}\Big[u_k(2h,2h)+\frac{1}{2}(u_k(2h,h)+u_k(h,2h)+u_k(3h,2h)+u_k(2h,3h)+u_k(h,h)+u_k(3h,3h))\Big].$$

对多重网格的每一层 k 定义如下内积：

$$(u_k,v_k)=\sum_{x\in\Omega_{kl}}h_k^d u_k(\boldsymbol{x})\overline{v_k(\boldsymbol{x})},$$

其中 d 是维数，$\Omega\in\mathbb{R}^d$. 从而可以定义**伴随映射**(adjoint injection)

$$(I_{k-1}^k u,v)_k=(u,(I_{k-1}^k)^* v)_{k-1},$$

称$(I_{k-1}^k)^*$为 I_{k-1}^k的伴随映射.

当 $d=2$ 时，易知

$$I_k^{k-1}=(I_{k-1}^k)^*=\frac{h^2}{(2h)^2}(\bar{I}_{k-1}^k)^{\mathrm{T}}=\frac{1}{4}(I_{k-1}^k)^{\mathrm{T}}.$$

因此，九点限制是九点延拓的伴随，七点限制是七点延拓的伴随.

关于延拓和限制的附注：

(1) 最显然的一种限制算子是直接映射(injection)，即粗网向量直接取相应细网点上的值，这种算子没有伴随.

(2) 令 $2m$ 为所求解的微分方程的阶，m_p 为延拓的阶，m_R 为限制的阶，则它们应满足条件 $m_p+m_R>2m$.

13.5 非线性多重网格法

非线性微分方程，如二元方程

$$-\Delta u(x,y)=\mathrm{e}^{u(x,y)},$$

其右端项关于 $u(x,y)$是非线性的，或边界条件是非线性的. 把微分方程和边界条件结合起来，可以得到抽象方程

$$\hat{L}u=f. \qquad (13.5.1)$$

前面所述的线性多重网格方法完全适合于这类非线性边值问题，这里仅讨论两种方法：13.5.1 节讨论基于线性化的方法，13.5.2 节讨论非线性多重网格方法的本质处理.

13.5.1 牛顿多重网格法

牛顿多重网格法是一种基于线性化的方法. 令

$$\varphi(u) = \hat{L}u - f = 0. \tag{13.5.2}$$

在 $\Omega_k(k=0,1,2,\cdots,M)$ 上分别用五点差分离散上式后，得到如下离散形式的非线性方程组

$$\boldsymbol{\varphi}_k(\boldsymbol{u}_k) = \boldsymbol{0} \quad (k = 0,1,2,\cdots,M), \tag{13.5.3}$$

其中 $\boldsymbol{\varphi}_k, \boldsymbol{u}_k \in \mathbb{R}^{N_k}$, $N_k = I_k \times J_k$, I_k 为 x 方向的网点数，J_k 为 y 方向的网点数.

将 $\boldsymbol{\varphi}_k(\boldsymbol{u}_k)$ 在 $\boldsymbol{u}_k^{(n)}$ 处作泰勒展开，舍去 $\boldsymbol{u}_k - \boldsymbol{u}_k^{(n)}$ 的高阶项，留下线性项，这本质上是把 $\boldsymbol{\varphi}_k(\boldsymbol{u}_k)$ 线性化. 得到如下的牛顿迭代公式：

$$\begin{cases} \boldsymbol{\psi}_k(\boldsymbol{u}_k^{(n)})\boldsymbol{w}_k = \boldsymbol{d}_k^{(n)}, \\ \boldsymbol{u}_k^{(n+1)} = \boldsymbol{u}_k^{(n)} + \boldsymbol{w}_k, \end{cases} \quad n = 0,1,2,\cdots, \tag{13.5.4}$$

其中

$$\boldsymbol{d}_k^{(n)} = -\boldsymbol{\varphi}_k(\boldsymbol{u}_k^{(n)}),$$

$$\boldsymbol{\psi}_k(\boldsymbol{u}_k) = \frac{\partial \boldsymbol{\varphi}_k}{\partial \boldsymbol{u}_k}(\boldsymbol{u}_k) = \begin{bmatrix} \dfrac{\partial \varphi_k^1}{\partial u_k^1} & \dfrac{\partial \varphi_k^1}{\partial u_k^2} & \cdots & \dfrac{\partial \varphi_k^1}{\partial u_k^n} \\ \dfrac{\partial \varphi_k^2}{\partial u_k^1} & \dfrac{\partial \varphi_k^2}{\partial u_k^2} & \cdots & \dfrac{\partial \varphi_k^2}{\partial u_k^n} \\ \vdots & \vdots & & \vdots \\ \dfrac{\partial \varphi_k^n}{\partial u_k^1} & \dfrac{\partial \varphi_k^n}{\partial u_k^2} & \cdots & \dfrac{\partial \varphi_k^n}{\partial u_k^n} \end{bmatrix} (\boldsymbol{u}_k)$$

这是向量 $\boldsymbol{\varphi}_k$ 对向量 $\boldsymbol{u}_k$ 的导数，即在 $\boldsymbol{u}_k$ 处的雅可比矩阵.

牛顿多重网格算法如下：

(1) 给定初值 $u_k^{(0)}$.

(2) 进行迭代,$n=0,1,2,\cdots$.

① 计算 $d_k^{(n)}=-\varphi_k(u_k^{(n)}),\psi_k^{(n)}=\dfrac{\partial\varphi_k}{\partial u_k}(u_k^{(n)})$.

② 以 $w_k^{(0)}$ 为初值,对 $\psi_k^{(n)}w_k=d_k^{(n)}$ 执行一个线性多重网格迭代步,得到 $w_k^{(1)}$.

③ 计算 $u_k^{(n+1)}=u_k^{(n)}+w_k^{(1)}$.

这是一个典型的牛顿多重网格迭代算法.

13.5.2 非线性多重网格法

利用非线性多重网格算法能避免牛顿迭代中的线性化,这个算法与线性多重网格法一样有效.下面讨论这个算法,注意它与线性情况的不同之处.

1. 非线性阻尼雅可比迭代方法

在多重网格法中,我们感兴趣的是非线性迭代法对高频误差分量的光滑性质.众所周知,在迭代方法中引入松弛参数 ω,当 $0<\omega<1$ 时,低松弛校正可以扩大收敛域,如将牛顿迭代法(13.5.4)的第二式改为 $u_k^{(n+1)}=u_k^{(n)}+\omega w_k$.本节着重讨论雅可比-牛顿 ω 松弛法和雅可比-皮卡(Picard)ω 松弛法.

考虑非线性问题(13.5.1)的离散方程组

$$\hat{L}_h u_h = f_h,$$

为讨论方便起见,将该非线性差分方程写成如下形式:

$$\hat{L}_h u_h = L_h u_h + g(x,u_h) = f_h,$$

其中 L_h 为线性差分算子.为讨论方便起见,设 L_h 为五点差分算子,$g(x,u_h)$为 u_h 的线性或非线性函数.

类似于式(13.4.7)和式(13.4.8),一个非线性阻尼雅可比松弛定义为

$$\bar{u}_h^{(n)} = u_h^{(n)} + \omega(z_h - u_h^{(n)}), \qquad (13.5.5)$$

$$\frac{4}{h^2}z_h + \left(L_h u_h^{(n)} - \frac{4}{h^2}u_h^{(n)}\right) + g(x, z_h) = f_h. \quad (13.5.6)$$

将式(13.5.6)中的非线性项 $g(x, z_h)$ 在 $u_h^{(n)}$ 处作泰勒展开，仅留下关于$(z_h - u_h^{(n)})$的线性项，则被线性化为

$$\frac{4}{h^2}z_h + \left(L_h u_h^{(n)} - \frac{4}{h^2}u_h^{(n)}\right) + g(x, u_h^{(n)}) + \frac{\partial g}{\partial u}(x, u_h^{(n)})(z_h - u_h^{(n)}) = f_h, \quad (13.5.7)$$

称式(13.5.5)和式(13.5.7)为雅可比-牛顿 ω 松弛法.

若用 $g(x, u_h^{(n)})$代替式(13.5.6)中的 $g(x, z_h)$，则有

$$\frac{4}{h^2}z_h + \left(L_h u_h^{(n)} - \frac{4}{h^2}u_h^{(n)}\right) + g(x, u_h^{(n)}) = f_h, \quad (13.5.8)$$

称式(13.5.5)和式(13.5.8)为雅可比-皮卡 ω 松弛法.

为具体讨论以上两种迭代方法的光滑效应，假设 g 为 u 的线性函数，即

$$g(x, u) = cu, \quad c > 0,$$

显然$\frac{\partial g}{\partial u} = c$，将它们代入式(13.5.6)和式(13.5.7)，则雅可比-牛顿 ω 松弛法的迭代矩阵为

$$B_{JN\omega} = \left(1 - \frac{\omega_N ch^2}{4 + ch^2}\right)I_h - \frac{\omega_N h^2}{4 + ch^2}L_h, \quad (13.5.9)$$

雅可比-皮卡 ω 松弛法的迭代矩阵为

$$B_{JP\omega} = \left(1 - \frac{\omega_P ch^2}{4}\right)I_h - \frac{\omega_P h^2}{4}L_h. \quad (13.5.10)$$

当 $\omega_N = \frac{4 + ch^2}{4}\omega_P$ 时，$B_{JN\omega} = B_{JP\omega}$. 由式(13.4.9)和式(13.4.10)可知

$$B_{J\omega} = I_h - \frac{\omega h^2}{4}L_h, \quad (13.5.11)$$

$$\lambda(\boldsymbol{B}_{\mathrm{J}\omega}) = 1 - \frac{\omega}{2}\left(2 - \cos\frac{k_1\pi}{N} - \cos\frac{k_2\pi}{N}\right), \quad 1 \leqslant k_1, k_2 \leqslant N-1, \tag{13.5.12}$$

将式(13.5.11)和式(13.5.12)代入式(13.5.9)和式(13.5.10),得

$$\boldsymbol{B}_{\mathrm{JN}\omega} = \frac{4}{4+ch^2}\boldsymbol{B}_{\mathrm{J}\omega} + \frac{(1-\omega_{\mathrm{N}})ch^2}{4+ch^2},$$

$$\boldsymbol{B}_{\mathrm{JP}\omega} = \boldsymbol{B}_{\mathrm{J}\omega} - \frac{\omega_{\mathrm{P}}ch^2}{4}.$$

易得它们的特征值分别为

$$\lambda(\boldsymbol{B}_{\mathrm{JN}\omega}) = \frac{4}{4+ch^2}\lambda(\boldsymbol{B}_{\mathrm{J}\omega}) + \frac{(1-\omega_{\mathrm{N}})ch^2}{4+ch^2}$$

$$\lambda(\boldsymbol{B}_{\mathrm{JP}\omega}) = \lambda(\boldsymbol{B}_{\mathrm{J}\omega}) - \frac{\omega_{\mathrm{P}}ch^2}{4}$$

雅可比-牛顿 ω 松弛法的光滑因子为

$$\begin{aligned}\mu(\boldsymbol{B}_{\mathrm{JN}\omega}) &= \max_{N/2\leqslant|\boldsymbol{k}|\leqslant N-1} |\lambda(\boldsymbol{B}_{\mathrm{JN}\omega})| \\ &= \max\left\{\left|1-\omega_{\mathrm{N}}\left(1-\frac{2\cos(\pi h)}{4+ch^2}\right)\right|, \left|1-\omega_{\mathrm{N}}\left(1+\frac{4\cos(\pi h)}{4+ch^2}\right)\right|\right\},\end{aligned} \tag{13.5.13}$$

其中 $\boldsymbol{k}=(k_1,k_2)$, $|\boldsymbol{k}|=\max\{k_1,k_2\}$. 上式括弧内的两个值,它们是 k_1 和 k_2 分别取 $N/2$ 和 $N-1$ 时得到的 $\lambda(\boldsymbol{B}_{\mathrm{JN}\omega})$ 以及 k_1 和 k_2 均取 $N-1$ 时得到的 $\lambda(\boldsymbol{B}_{\mathrm{JN}\omega})$. 从上式可知对固定的 $\omega_{\mathrm{N}}(0<\omega_{\mathrm{N}}<1)$ 和固定的 h,当 $c\to\infty$ 时,有

$$\lim_{c\to\infty}\mu(\boldsymbol{B}_{\mathrm{JN}\omega}) = |1-\omega_{\mathrm{N}}|.$$

可以求出雅可比-牛顿 ω 松弛法的最佳松弛因子和最佳光滑因子分别为

$$\omega_{\mathrm{opt}} = \frac{4+ch^2}{4+ch^2+\cos(\pi h)}, \quad \mu_{\mathrm{opt}} = \frac{3\cos(\pi h)}{4+ch^2+\cos(\pi h)}.$$

易见对固定的 h,雅可比-牛顿 ω 松弛法的光滑因子随 c 的增加而变好.

对雅可比-皮卡 ω 松弛法，将 $\omega_N=\frac{4+ch^2}{4}\omega_P$ 代入式(13.5.13)，则有雅可比-皮卡 ω 松弛法的光滑因子

$$\mu(B_{JP_\omega})=\max\left\{\left|1-\omega_P\frac{4+ch^2}{4}\left(1-\frac{2\cos(\pi h)}{4+ch^2}\right)\right|,\right.$$

$$\left.\left|1-\omega_P\frac{4+ch^2}{4}\left(1+\frac{4\cos(\pi h)}{4+ch^2}\right)\right|\right\}$$

$$=\max\left\{\left|1-\omega_P\left(1+\frac{ch^2-2\cos(\pi h)}{4}\right)\right|,\right.$$

$$\left.\left|1-\omega_P\frac{4+ch^2+4\cos(\pi h)}{4}\right|\right\}.$$

易知当固定 $\omega_P(0<\omega_P<1)$ 和 h，随 c 增加到充分大时，雅可比-皮卡 ω 松弛法没有光滑性质.

2. 亏损方程

考虑非线性问题(13.5.1)的离散方程组

$$\hat{L}_h u_h=f_h,$$

其中 $\hat{L}_h$ 为差分算子，h 为网格步长. 令 u_h^* 为精确解，u_h 为近似解，v_h 为误差. 将 $u_h^*=u_h+v_h$ 代入上式，有

$$\hat{L}_h(u_h+v_h)=f_h.$$

当 $\hat{L}_h$ 为线性算子 L_h 时，则

$$L_h u_h+L_h v_h=f_h,$$

亏损方程为

$$L_h v_h=d_h,$$

其中 $d_h=f_h-L_h u_h$. 从亏损方程求出 v_h，用它对 u_h 进行修正.

当 $\hat{L}_h$ 为非线性算子时，$\hat{L}_h$ 不能分别作用于 u_h 和 v_h，则

$$\hat{L}_h(u_h+v_h)-\hat{L}_h u_h=f_h-\hat{L}_h u_h.$$

仍令 $d_h=f_h-\hat{L}_h u_h$，则亏损方程为

$$\hat{L}_h(u_h+v_h)-\hat{L}_h u_h=d_h,$$

在粗网 Ω_H 上的亏损方程为

$$\hat{L}_H(I_h^H u_h + v_H) - \hat{L}_H(I_h^H u_h) = I_h^H d_h,$$

其中 I_h^H 为限制算子. 与线性情况不同,不是在粗网上直接求解校正量v_H,而是求解完全逼近量 $u_H = I_h^H u_h + v_H$,即求解方程

$$\hat{L}_H u_H = d_H + \hat{L}_H(I_h^H u_h),$$

然后,有校正量

$$v_H = u_H - I_h^H u_h.$$

3. FAS 双网格方法

FAS(full approximation storage)**方法**是非线性多重网格法,它的特点是不存储校正量v_H,而是存储完全近似量 $u_H = I_h^H u_h + v_H$. 其中用 G^ν 表示非线性松弛法松弛 ν 次的松弛算子,而不同于线性松弛法中的松弛算子符号 S^ν.

FAS 双网格算法为:

(1) 前光滑:对 $u_h^{(n)}$ 在 Ω_h 上作 ν_1 次非线性松弛,得到

$$\bar{u}_h^{(n)} \stackrel{\text{def}}{=} G^{\nu_1}(u_h^{(n)}, \hat{L}_h, f_h).$$

(2) 粗网修正:

① 限制近似解:$\bar{u}_H^{(n)} \stackrel{\text{def}}{=} I_h^H \bar{u}_h^{(n)}$.

② 计算网格亏损量:$d_h^{(n)} = f_h - L_h u_h^{(n)}$.

③ 限制亏损量:$d_H^{(n)} = I_h^H d_h^{(n)}$.

④ 计算 $f_H = d_H + \hat{L}_H \bar{u}_H^{(n)}$,在 Ω_H 上求解亏损方程$\hat{L}_H u_H = f_H$,解得 $\bar{u}_H$.

⑤ 计算 Ω_H 上的修正量:$v_H^{(n)} = \bar{u}_H - I_h^H u_h^{(n)}$.

⑥ 插值修正量:$v_h^{(n)} = I_H^h v_H^{(n)}$.

⑦ 对 $\bar{u}_h^{(n)}$ 作修正:$\hat{u}_h^{(n)} = \bar{u}_h^{(n)} + v_h^{(n)}$.

(3) 后光滑:对 $\hat{u}_h^{(n)}$ 在 Ω_h 上作 ν_2 次非线性松弛,得到

$$u_h^{(n+1)} := G^{\nu_2}(\hat{u}_h^{(n)}, \hat{L}_h, f_h).$$

4. FAS 多重网格法

下面讨论用 FAS 多重网格法求解 Ω_M 上离散的非线性方

程组

$$\hat{L}_M u_M = f_M$$

的计算过程.已知 $u_M^{(n)}$,求 $u_M^{(n+1)} := \mathrm{MFAS}(u_M^{(n)}, \hat{L}_M, f_M)$.仿 FAS 双网格法,计算分三个阶段:

(1) 前光滑:对 $u_M^{(n)}$ 在 Ω_M 上作 ν_1 次非线性松弛,得到

$$\bar{u}_M^{(n)} := G^{\nu_1}(u_M^{(n)}, \hat{L}_M, f_M).$$

(2) 粗网修正:

① 限制近似解:$\bar{u}_{M-1}^{(n)} := I_M^{M-1}\bar{u}_M^{(n)}$.

② 计算 Ω_M 上的亏损量:$d_M^{(n)} = f_M - L_M u_M^{(n)}$.

③ 限制亏损量:$d_{M-1}^{(n)} = I_M^{M-1} d_M^{(n)}$.

④ 计算 $f_{M-1} = d_{M-1} + \hat{L}_{M-1}\bar{u}_{M-1}^{(n)}$,在 Ω_{M-1} 上求解非线性亏损方程 $\hat{L}_{M-1} u_{M-1} = f_{M-1}$,即计算 $\bar{u}_{M-1} := \mathrm{MFAS}(\bar{u}_{M-1}^{(n)}, \hat{L}_{M-1}, f_{M-1})$,以 $\bar{u}_{M-1}^{(n)}$ 作为初始值,作 MFAS 网格法的 $r(r \geqslant 1)$ 型迭代,需用到网格 $\Omega_{M-1}, \Omega_{M-2}, \cdots, \Omega_0$.

⑤ 计算 Ω_{M-1} 上的修正量:$v_{M-1}^{(n)} = \bar{u}_{M-1} - I_M^{M-1}\bar{u}_M^{(n)}$.

⑥ 插值修正量:$v_M^{(n)} = I_{M-1}^M v_{M-1}^{(n)}$.

⑦ 对 $\bar{u}_M^{(n)}$ 作修正:$\hat{u}_M^{(n)} = \bar{u}_M^{(n)} + v_M^{(n)}$.

(3) 后光滑:对 $\hat{u}_M^{(n)}$ 在 Ω_M 上作 ν_2 次非线性松弛,得到

$$u_M^{(n+1)} := G^{\nu_2}(\hat{u}_M^{(n)}, \hat{L}_M, f_M).$$

13.6 计算机上的执行性能

13.6.1 数据结构

多重网格法的程序可以做到高度模块化,如松弛、插值和限制均可编成独立的子程序,容易调用、容易替换.除此之外,对于规则区域还需采用合理的数据结构,当然,对于不规则区域或网格局部加密情况除外.

对于规则区域，只需用两个数组. 一个数组$\boldsymbol{v}$存放每层网格上的解向量和误差向量，从最粗网格到最细网格，逐点依次排列；另一个数组$\boldsymbol{f}$存放每层网格上的右端项和余项，也是从最粗网格到最细网格依次排列. 这两个数组均应包括边界点，对一维问题，第k层有2^k+1个点.

图 13.21 描述了间隔数 $N=16$，网格层数 $M=4$，V 循环的一维问题的典型数据结构，说明数据结构随 V 循环在变化.

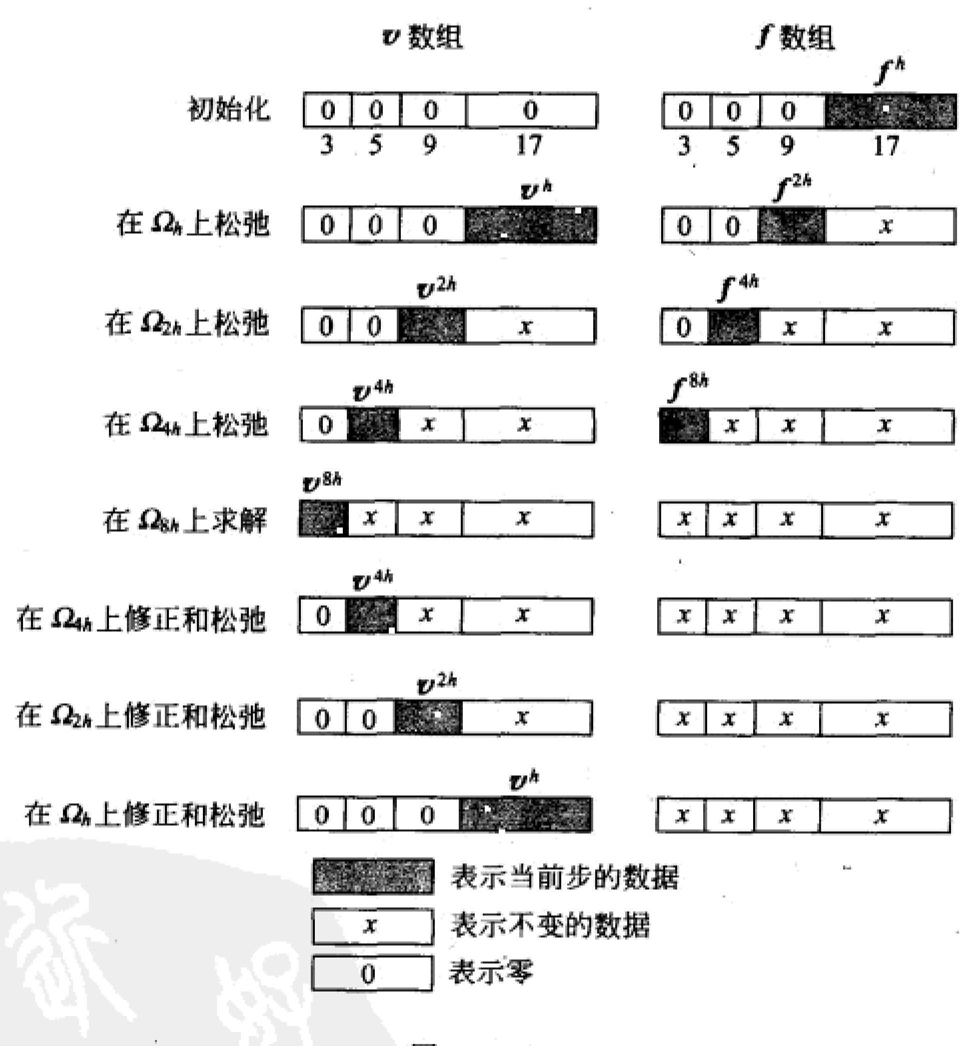


图 13.21

先初始化每层网格的解向量，即置数组$\boldsymbol{v}$为零，最细网格上的右边数组$\boldsymbol{f}$是已知的，其余层的$\boldsymbol{f}$也置为零. 当V循环往粗网方向下降时，每层网格上的松弛把该层解向量数组填满，同时把余项转移到下一层粗网，把该层的右端数组填满. 当V循环往细网方向上升时，右边数组不变，但在每层网格上又进行松弛，重写该层的解数组段，同时把前一层粗网上的解数组段冲成零，目的是为执行另一个V循环提供零初始值.

二维问题的数据结构和一维类似，仅仅每层网格上的数据段比一维情况长些.

13.6.2 存储量

考虑d维n层网格，每层网点为N^d，$N=2^n$. 每层存储$\boldsymbol{v}$和$\boldsymbol{f}$两个数组. 存储量

$$\begin{aligned}M_d &= 2N^d + 2(N/2)^d + \cdots + 2(N/2^n)^d \\ &= 2N^d(1 + 2^{-d} + 2^{-2d} + \cdots + 2^{-nd}) < 2N^d/(1-2^{-d}).\end{aligned}$$

当$d=1$时，$M_1<2(2N)$，不足2倍细网存储量.

当$d=2$时，$M_2<\frac{4}{3}(2N^2)$，不足4/3倍细网存储量.

当$d=3$时，$M_3<\frac{8}{7}(2N^3)$，不足8/7倍细网存储量.

因此，多重网格法的存储量随问题维数的增加而相对地下降.

13.6.3 计算量

记WU(work unit)为工作单位，即一个WU为细网上执行一次松弛所需的算术运算量，$\nu_1=\nu_2=1$的一个V循环多重网格法的工作量为

$$W_{Vd} = 2\mathrm{WU}(1 + 2^{-d} + 2^{-2d} + \cdots + 2^{-nd}) < \frac{2}{1-2^{-d}}\ \mathrm{WU}$$

则有

$$W_{V1} < 4\ \mathrm{WU},\quad W_{V2} < \frac{8}{3}\ \mathrm{WU},\quad W_{V3} < \frac{16}{7}\ \mathrm{WU}.$$

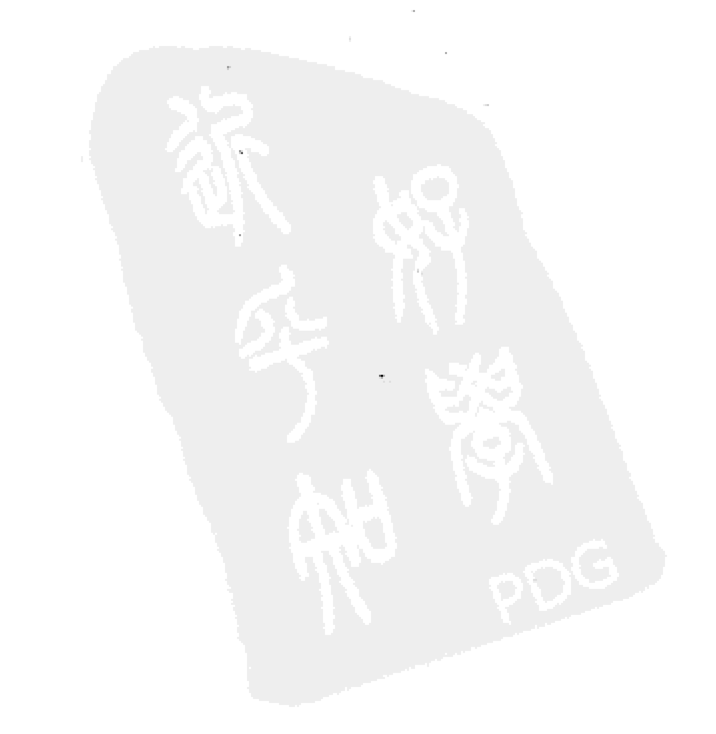
$\nu_1=\nu_2=1$ 的一个 FMV 多重网格法的工作量为

$$W_{\mathrm{FMV}d} = \frac{2\ \mathrm{WU}}{1-2^{-d}}(1+2^{-d}+2^{-2d}+\cdots+2^{-nd}) < \frac{2}{(1-2^{-d})^2}\ \mathrm{WU}$$

则有

$$W_{\mathrm{FMV1}} < 8\ \mathrm{WU},\quad W_{\mathrm{FMV2}} < \frac{7}{2}\ \mathrm{WU},\quad W_{\mathrm{FMV3}} < \frac{5}{2}\ \mathrm{WU}.$$

设 p 为微分方程的阶，M 是一个常数，ϵ 是一个小量为解的计算精度，则在 $h<(\epsilon/(2M))^{1/p}$ 条件下，多重网格法收敛到截断误差级所需的运算量为：V 循环多重网格法的总运算量为 $O(N^d\log N)$，FMV 多重网格法的总运算量为 $O(N^d)$. 这比求解线性代数方程组的直接方法的计算速度还要快.



14 积分方程数值解法

14.1 引言

积分方程(integral equation)

$$y(x)=\lambda\int_a^b K(x,s)y(s)\mathrm{d}s+f(x) \tag{14.1.1}$$

称为**第二类(弗雷德霍姆)(Fredholm)积分方程**(Fredholm integral equation of the second kind),其中 y 为未知函数,K 为**积分方程的核**(kernel of the integral equation),f 为右端项或称自由项,λ 为参数. λ,K 和 f 均为已知. 积分方程(14.1.1)的数值解法即是求未知函数 y 的近似解 $\tilde{y}$.

积分方程

$$\int_a^b K(x,s)y(s)\mathrm{d}s=\kappa y(x) \tag{14.1.2}$$

有非零解 y,此时 κ 为**核的特征值**(eigenvalue of the kernel),而 y 为对应于特征值 κ 的**特征函数**(eigenfunction). 方程(14.1.2)的数值解法即是求 κ 和 y 的近似解.

积分方程

$$y(x)=\lambda\int_0^x K(x,s)y(s)\mathrm{d}s+f(x) \tag{14.1.3}$$

称为**第二类(沃尔泰拉)积分方程**(Volterra integral equation of second kind). 其数值解法即为求近似解 $\tilde{y}$.

14.2 第二类弗雷德霍姆积分方程的数值求积方法

14.2.1 方法的一般描述

求解第二类弗雷德霍姆积分方程

$$y(x)=\lambda\int_a^b K(x,s)y(s)\mathrm{d}s+f(x)$$

的**数值求积分方法**(quadrature method)或称**有限和方法**(method of finite sums)是：取某一数值求积分公式

$$\int_a^b F(x)\mathrm{d}x\approx\sum_{k=1}^{n}A_kF(x_k),\tag{14.2.1}$$

其中 $x_k\in[a,b]$是求积分的节点，A_k 是不依赖于 F 的求积分公式的系数.

把第二类弗雷德霍姆积分方程中的定积分用数值求积分公式来代替，于是积分方程变为一个近似式

$$y(x)\approx\lambda\sum_{k=1}^{n}A_kK(x,x_k)y(x_k)+f(x).\tag{14.2.2}$$

如果把上式写为

$$\tilde{y}(x)=\lambda\sum_{k=1}^{n}A_kK(x,x_k)\,\tilde{y}(x_k)+f(x),\tag{14.2.3}$$

则$\tilde{y}$为 y 的近似解. 令 $x=x_1,x_2,\cdots,x_n$，并采用记号$\tilde{y}_k=\tilde{y}(x_k)$，$K_{ik}=K(x_i,x_k)$，$f_k=f(x_k)$，由式(14.2.3)得线性代数方程组

$$\tilde{y}_i=\lambda\sum_{k=1}^{n}A_kK_{ik}\,\tilde{y}_k+f_i\quad(i=1,2,\cdots,n).\tag{14.2.4}$$

如果方程组(14.2.4)的行列式

$$|\,\boldsymbol{I}-\lambda\boldsymbol{L}\,|\neq0,\tag{14.2.5}$$

那么方程组有惟一解$\tilde{y}_1,\tilde{y}_2,\cdots,\tilde{y}_n$，其中

$$L = \begin{pmatrix} A_1K_{11} & A_2K_{12} & \cdots & A_nK_{1n} \\ A_1K_{21} & A_2K_{22} & \cdots & A_nK_{2n} \\ \vdots & \vdots & & \vdots \\ A_1K_{n1} & A_2K_{n2} & \cdots & A_nK_{nn} \end{pmatrix},$$

$\tilde{y}_1, \tilde{y}_2, \cdots, \tilde{y}_n$ 就是积分方程(14.1.1)的解 y 在求积分公式的节点 $x_1, x_2, \cdots, x_n$ 上的近似值. 利用这些值可以求出任意 $x \in [a,b]$ 的积分方程(14.1.1)的近似解

$$\tilde{y}(x) = f(x) + \lambda \sum_{k=1}^{n} A_k K(x, x_k)\, \tilde{y}_k. \qquad (14.2.6)$$

例 14.2.1 用辛普森求积分公式和三点高斯-勒让德(Gauss-Legendre)求积分公式并解积分方程

$$y(x) = \int_0^1 (1 - 3xs) y(s) \mathrm{d}s + (1 - 3x).$$

解 辛普森求积公式为

$$\int_0^1 F(x) \mathrm{d}x \approx \frac{1}{6}\left[F(0) + 4F\left(\frac{1}{2}\right) + F(1)\right],$$

应用此公式到积分方程得线性代数方程组

$$\begin{cases} \tilde{y}_1 = \dfrac{1}{6}[\tilde{y}_1 + 4\,\tilde{y}_2 + \tilde{y}_3] + 1, \\ \tilde{y}_2 = \dfrac{1}{6}[\tilde{y}_1 + \tilde{y}_2 - \dfrac{1}{2}\,\tilde{y}_3] - \dfrac{1}{2}, \\ \tilde{y}_3 = \dfrac{1}{6}[\tilde{y}_1 - 2\,\tilde{y}_2 - 2\,\tilde{y}_3] - 2. \end{cases}$$

解这个方程组,得 $\tilde{y}_1 = \dfrac{2}{3}, \tilde{y}_2 = -\dfrac{1}{3}, \tilde{y}_3 = -\dfrac{4}{3}$.

利用近似解表达式(14.2.6)可以得出

$$\tilde{y}\left(\frac{1}{4}\right) = \frac{1}{6}, \quad \tilde{y}\left(\frac{3}{4}\right) = -\frac{5}{6}.$$

三点高斯-勒让德求积公式为

$$\int_0^1 F(x)\mathrm{d}x \approx \frac{1}{18}\left[5F(\alpha)+8F\left(\frac{1}{2}\right)+5F(\beta)\right],$$

其中

$$\alpha=\frac{1}{2}(1-\sqrt{0.6})=0.112702,$$

$$\beta=\frac{1}{2}(1+\sqrt{0.6})=0.887298,$$

应用此积分公式到积分方程得线性代数方程组

$$\begin{cases}0.732807\,\tilde{y}_1-0.369310\,\tilde{y}_2-0.194444\,\tilde{y}_3=0.661895,\\ -0.230819\,\tilde{y}_1+0.888889\,\tilde{y}_2+0.091930\,\tilde{y}_3=-0.5,\\ -0.194444\,\tilde{y}_1+0.147088\,\tilde{y}_2+1.378304\,\tilde{y}_3=-1.661894.\end{cases}$$

解之,得

$$\tilde{y}_1=0.441264,\quad \tilde{y}_2=-0.333333,\quad \tilde{y}_3=-1.107929.$$

利用近似解表达式可得

$$\tilde{y}(0)=0.666667,\qquad \tilde{y}\left(\frac{1}{4}\right)=0.166667,$$

$$\tilde{y}\left(\frac{3}{4}\right)=-0.833334,\quad \tilde{y}(1)=-1.333333.$$

本例的解析解为 $y(x)=\frac{2}{3}(1-3x)$,可以看出求解是很精确的.

由于在求积分方程的近似解的过程中都要解线性代数方程组,而方程的个数与数值求积分公式的节点数相同,因此不宜靠增加节点数来提高求积公式的精度,故采用高斯求积分公式是较为合适的.

14.2.2 乘积积分法

在积分方程(14.1.1)中要计算积分

$$\int_a^b K(x,s)y(s)\mathrm{d}s.$$

如果固定 x,并令

$$W(s)=K(x,s),\quad s\in[a,b],$$

那么上述积分就化为通常的乘积积分

$$\int_a^b W(s)y(s)\mathrm{d}s.$$

由此可以采用**乘积积分的方法**(method of product integration)来计算.

在区间$[a,b]$上取 n 个不同的节点(通常取等距节点)s_1, $s_2,\cdots,s_n$,构造求积分公式

$$\int_a^b K(x,s)\varphi(s)\mathrm{d}s\approx\sum_{k=1}^n A_k(x)\varphi_k,$$

其中系数依赖于 x. 选取 $A_k(x)$使求积分公式的代数精度为 n,即要求对 $\varphi(s)=s^l(l=0,1,\cdots,n-1)$有

$$\sum_{k=1}^n A_k(x)s_k^l=\int_a^b K(x,s)s^l\mathrm{d}s\quad(l=0,1,\cdots,n-1),\tag{14.2.7}$$

这是一个线性代数方程组,其中

$$A_1(x),A_2(x),\cdots,A_n(x)$$

是未知函数. 当 $A_k(x)(k=1,2,\cdots,n)$求出之后,可以用近似式

$$y(x)\approx\lambda\sum_{k=1}^n A_k(x)y_k+f(x)$$

来代替积分方程(14.1.1).

如果令 $x=s_1,s_2,\cdots,s_n$,则得线性代数方程组

$$\tilde{y}_i=\lambda\sum_{k=1}^n A_k(s_i)\,\tilde{y}_k+f(s_i)\quad(i=1,2,\cdots,n),\tag{14.2.8}$$

由此可以求出解的近似值$\tilde{y}_1,\tilde{y}_2,\cdots,\tilde{y}_n$.

为了计算 $A_k(s_i)(k,i=1,2,\cdots,n)$,一般要求核 $K(x,s)$有解析表达式.

例 14.2.2 用乘积积分法解积分方程

$$y(x)=-\frac{1}{\pi}\int_{-1}^{1}\frac{y(s)}{1+(x-s)^2}\mathrm{d}s+1.$$

解 采用 11 个等距节点，得出 11 个未知数的 11 个方程所构成的方程组，解之得

$$\tilde{y}(0)=0.65741,\qquad \tilde{y}(\pm 0.2)=0.66151,$$
$$\tilde{y}(\pm 0.4)=0.67389,\quad \tilde{y}(\pm 0.6)=0.69448,$$
$$\tilde{y}(\pm 0.8)=0.72249,\quad \tilde{y}(\pm 1)=0.75572.$$

这个结果是相当精确的.

乘积积分方法可以用来求解其核 $K(x,y)$ 为奇异的情况，其奇异性的含义是核的一阶导数或多阶导数在积分区域内无定义的情况. 例如，

$$K(x,s)=\ln|x-s|\,\widetilde{K}(x,s),$$

其中 $\widetilde{K}(x,s)$ 是其变量的光滑函数.

14.2.3 修正的数值求积方法

使用数值求积方法时，已假定核函数 K 是充分光滑的. 如果 K 有奇异性质，则要进行一些辅助工作以抵消奇异性质的影响. 而乘积积分方法可以用于求解具有奇异性质的核 K 的积分方程.

如果当 $x=s$ 时，K 具有奇异性质，则先把积分方程(14.1.1)改写为

$$y(x)-\lambda y(x)\int_a^x K(x,s)\mathrm{d}s$$
$$=\lambda\int_a^x K(x,s)[y(s)-y(x)]\mathrm{d}s+f(x). \qquad (14.2.9)$$

等号左边的积分是不含未知函数的，因而可以计算出来. 而等号右边的积分，由于当 $x=s$ 时，有$[y(s)-y(x)]=0$，因此可以消除或减弱奇异性质.

利用数值求积公式(14.2.1)，并令$\tilde{y}$为积分方程的近似解，则可写出

$$\tilde{y}(x)[1-\lambda B(x)] = \lambda \sum_{k=1}^{n} A_k K(x,s_k)[\tilde{y}(s_k) - \tilde{y}(x)] + f(x),$$

其中

$$B(x) = \int_a^b K(x,s)\mathrm{d}s.$$

如果记

$$\Delta(x) = \sum_{k=1}^{n} A_k K(x,s_k) - B(x),$$

并令 $x=s_j$，就得到

$$\tilde{y}(s_j)[1+\lambda\Delta(s_j)] = \lambda \sum_{k=1}^{n} A_k K(s_j,s_k)\, \tilde{y}(s_k) + f(s_j)$$

$$j = 1,2,\cdots,n. \qquad (14.2.10)$$

公式(14.2.10)称为**修正的数值求积方法**(modified quadrature method).

例 14.2.3 求积分方程

$$y(x) = -\int_0^1 K(x,s)y(s)\mathrm{d}s + x^2$$

的近似解，其中

$$K(x,s) = \begin{cases} x(1-s), & \text{当 } 0 \leqslant x \leqslant s \leqslant 1, \\ s(1-x), & \text{当 } 0 \leqslant s \leqslant x \leqslant 1. \end{cases}$$

积分方程的精确解为 $y(x)=2\cosh x - 0.07335\sinh x - 2$.

解 先用一般数值求积方法来求解. 利用两个节点的高斯-勒让德求积分公式，节点为 $s_1=0.21132$，$s_2=0.78868$. 相应的求积系数为 $A_1=A_2=\frac{1}{2}$，那么可以求得线性代数方程组

$$\begin{cases} 1.08333\,\tilde{y}_1 + 0.02233\,\tilde{y}_2 = 0.04466, \\ 0.02233\,\tilde{y}_1 + 1.08333\,\tilde{y}_2 = 0.62202. \end{cases}$$

解此方程组，得其解

$$\tilde{y}_1 = 0.02940, \quad \tilde{y}_2 = 0.57357.$$

将此解与积分方程的解在相应的节点上的值相比较有

$$y(s_1)-\tilde{y}_1=-0.00020,\quad y(s_2)-\tilde{y}_2=0.01732.$$

由于此例中积分方程的核 K 在 $x=s$ 处一阶导数不连续，为此应采用修正的数值求积方法，把积分方程写为式(14.2.9)的形式($\lambda=-1$)：

$$y(x)\left[1+\int_0^1 K(x,s)\mathrm{d}s\right]=-\int_0^1 K(x,s)[y(s)-y(x)]\mathrm{d}s+x^2.$$

由于

$$\int_0^1 K(x,s)\mathrm{d}s=\frac{x}{2}(1-x),$$

上面的积分方程可以写为

$$y(x)\left[1+\frac{x}{2}(1-x)\right]+\int_0^1 K(x,s)[y(x)-y(s)]\mathrm{d}s=x^2.$$

仍利用两个节点高斯-勒让德求积公式代替上述积分，并令 $x=s_1,s_2$，就得到线性方程组

$$\begin{cases}1.06100\,\tilde{y}_1+0.02233\,\tilde{y}_2=0.04466,\\0.02233\,\tilde{y}_1+1.06100\,\tilde{y}_2=0.62202.\end{cases}$$

解这个方程组，得

$$\tilde{y}_1=0.02976,\quad \tilde{y}_2=0.58564.$$

再用积分方程的精确解 $y(x)$ 在求积节点上的值与其比较，可以得到

$$y(s_1)-\tilde{y}_1=-0.00056,\quad y(s_2)-\tilde{y}_2=0.00525.$$

由此可以看出，采用修正的数值求积方法得到的数值结果更精确.

14.2.4 重叠核方法

当积分方程的核 K 有奇点，而又能够算出**重叠核**(iterated kernel)

$$K_2(x,s)=\int_a^b K(x,t)K(t,s)\mathrm{d}t$$

时,可以把积分方程(14.1.1)化为如下形式:

$$y(x)=\lambda^2\int_a^b K_2(x,s)y(s)\mathrm{d}s+\lambda\int_a^b K(x,s)f(s)\mathrm{d}s+f(x), \tag{14.2.11}$$

其中积分

$$\int_a^b K(x,s)f(s)\mathrm{d}x$$

不包含未知函数,因而可以计算出来.而含重叠核 K_2 的积分用数值求积会比含 K 的积分更有效,这个方法称为**重叠核方法**(iterated kernel method).

例 14.2.4 用重叠核方法求例 14.2.3 中积分方程的近似解.

解 积分方程的重叠核为

$$K_2(x,s)=\begin{cases}\dfrac{1}{6}(xs^3+sx^3-3xs^2+2xs-x^3), & 0\leqslant x\leqslant s\leqslant 1,\\ \dfrac{1}{6}(sx^3+xs^3-3sx^2+2sx-s^3), & 0\leqslant s\leqslant x\leqslant 1.\end{cases}$$

此例中,$\lambda=-1$,$f(x)=x^2$,并且

$$\lambda\int_0^1 K(x,s)f(s)\mathrm{d}s+f(x)=x^2-\frac{1}{12}x+\frac{1}{12}x^4.$$

记等式右边部分为 $F(x)$,则相应于式(14.2.9)有

$$y(x)=\int_0^1 K_2(x,s)y(s)\mathrm{d}s+F(x).$$

对此积分方程应用数值求积方法,仍采用两个节点的高斯-勒让德求积公式,有

$$K_2(s_1,s_1)=K_2(s_2,s_2)=0.00926,$$
$$K_2(s_1,s_2)=K_2(s_2,s_1)=0.00678,$$
$$F(s_1)=0.02722,\quad F(s_2)=0.58854.$$

由此得线性代数方程组

$$\begin{cases}0.99537\,\tilde{y}_1 - 0.00339\,\tilde{y}_2 = 0.02722,\\ -0.00339\,\tilde{y}_1 + 0.99537\,\tilde{y}_2 = 0.58854.\end{cases}$$

解这个方程组，得出

$$\tilde{y}_1 = 0.02936,\quad \tilde{y}_2 = 0.57138,$$

此解与积分方程的精确解 y 在求积的节点上的值相比较有

$$y(s_1) - \tilde{y}_1 = -0.00016,\quad y(s_2) - \tilde{y}_2 = 0.00049.$$

可以看出，这是相当精确的.

14.3 近似退化核替代法

对于第二类弗雷德霍姆积分方程(14.1.1)：

$$y(x) = \lambda\int_a^b K(x,s)y(s)\mathrm{d}s + f(x),$$

如果其核 K 可以表示为

$$K(x,s) = \sum_{i=1}^{n}\alpha_i(x)\beta_i(s),$$

则称其核是**退化的**，其中假定函数系 $\{\alpha_i(x)\}$ 是线性无关的，$\{\beta_i(s)\}$ 也是线性无关的.

近似退化核替代法(method of replacing approximately the kernel by a degenerate one)是基于退化核积分方程求其精确解的方法. 设积分方程(14.1.1)的核 K 可以近似地用一个退化核来替代，即

$$K(x,s) \approx \sum_{i=1}^{n}\alpha_i(x)\beta_i(s), \tag{14.3.1}$$

并设寻求的方程(14.1.1)的近似解 $\tilde{y}$ 形如

$$\tilde{y}(x) = \lambda\sum_{i=1}^{n}c_i\alpha_i(x) + f(x), \tag{14.3.2}$$

其中

$$c_i = \int_a^b \beta_i(s)\, \tilde{y}(s) \mathrm{d}s. \qquad (14.3.3)$$

把表达式(14.3.2)代入式(14.3.3),得到

$$c_i = \int_a^b \beta_i(s) f(s) \mathrm{d}s + \lambda \int_a^b \beta_i(s) \sum_{j=1}^{n} c_j \alpha_j(s) \mathrm{d}s \quad (i = 1,2,\cdots,n).$$

引入记号

$$f_i = \int_a^b \beta_i(s) f(s) \mathrm{d}s, \quad A_{ij} = \int_a^b \alpha_j(s) \beta_i(s) \mathrm{d}s, \quad (14.3.4)$$

从而得出

$$c_i - \lambda \sum_{j=1}^{n} c_j A_{ij} = f_i \quad (i = 1,2,\cdots,n). \qquad (14.3.5)$$

这是关于 c_i 的线性代数方程组,解此方程组就可得到 $c_1, c_2, \cdots, c_n$,把这些数代入表达式(14.3.2)就得到第二类弗雷德霍姆积分方程(14.1.1)的近似解.

通常,取 K 的按某一标准正交函数系 $\{\varphi_k\}$ 的傅里叶级数的部分和或 K 的泰勒级数的部分和作为近似退化核.

例 14.3.1 用近似退化核替代法求积分方程

$$y(x) = \int_0^1 \sinh(xs) y(s) \mathrm{d}s + 1 - x^2$$

的近似解.

解 对积分方程的核 $K(x,s) = \sinh(xs)$,用其 Taylor 级数的前三项的部分和

$$\sinh(xs) \approx xs + \frac{(xs)^3}{3!} + \frac{(xs)^5}{5!}$$

来替代,可取

$$\alpha_1(x) = x, \qquad \alpha_2(x) = x^3, \qquad \alpha_3(x) = x^5,$$

$$\beta_1(s) = s, \qquad \beta_2(s) = \frac{1}{3!} s^3, \qquad \beta_3(s) = \frac{1}{5!} s^5,$$

$$f(x) = 1 - x^2.$$

把近似解$\tilde{y}$取为式(14.3.2)的形式：

$$\tilde{y}(x) = c_1 x + c_2 x^3 + c_3 x^5 + 1 - x^2,$$

利用公式(14.3.4)计算出线性代数方程组的系数及右端项，有

$$f_1 = \int_0^1 \beta_1(s) f(s) \mathrm{d}s = \frac{1}{4},$$

$$f_2 = \int_0^1 \beta_2(s) f(s) \mathrm{d}s = \frac{1}{72},$$

$$f_3 = \int_0^1 \beta_3(s) f(s) \mathrm{d}s = \frac{1}{2880},$$

$$A_{11} = \int_0^1 s^2 \mathrm{d}s = \frac{1}{3}, \qquad A_{12} = \int_0^1 s^4 \mathrm{d}s = \frac{1}{5},$$

$$A_{13} = \int_0^1 s^6 \mathrm{d}s = \frac{1}{7}, \qquad A_{21} = \int_0^1 \frac{1}{3!} s^4 \mathrm{d}s = \frac{1}{30},$$

$$A_{22} = \int_0^1 \frac{1}{3!} s^6 \mathrm{d}s = \frac{1}{42}, \quad A_{23} = \int_0^1 \frac{1}{3!} s^8 \mathrm{d}s = \frac{1}{54},$$

$$A_{31} = \int_0^1 \frac{1}{5!} s^6 \mathrm{d}s = \frac{1}{840}, \quad A_{32} = \int_0^1 \frac{1}{5!} s^8 \mathrm{d}s = \frac{1}{1080},$$

$$A_{33} = \int_0^1 \frac{s^{10}}{5!} \mathrm{d}s = \frac{1}{1320}.$$

于是得到线性代数方程组

$$\begin{cases} c_1 = \dfrac{1}{3} c_1 + \dfrac{1}{5} c_2 + \dfrac{1}{7} c_3 + \dfrac{1}{4}, \\ c_2 = \dfrac{1}{30} c_1 + \dfrac{1}{42} c_2 + \dfrac{1}{54} c_3 + \dfrac{1}{72}, \\ c_3 = \dfrac{1}{840} c_1 + \dfrac{1}{1080} c_2 + \dfrac{1}{1320} c_3 + \dfrac{1}{2880}. \end{cases}$$

用迭代法求解，可得 $c_1 = 0.3833, c_2 = 0.0273, c_3 = 0.0008$，由此得近似解$\bar{y}(x)$：

$$\bar{y}(x) = 0.3833x + 0.0273x^2 + 0.0008x^5 + 1 - x^2.$$

14.4 基于解的展开方法

对于积分方程(14.1.1)的解 $y(x)$,如果写成

$$y(x) \approx y_N(x) = \sum_{i=1}^{N} a_i^{(N)} h_i(x), \tag{14.4.1}$$

其中$\{h_i(x)\}$是在 $\mathscr{L}^2(a,b)$内完备的.那么对于充分大的 N 以及 $a_i^{(N)}$ 的某种选择,可以得到方程(14.1.1)解 $y(x)$的相当好的近似 $y_N(x)$.

14.4.1 最小二乘近似

根据一般的最小二乘近似(least squares approximation)原理,对于固定的 N,令

$$I(\boldsymbol{a}^{(N)}) = \int_a^b [y_N(x) - f(x) - \int_a^b K(x,s) y_N(s) \mathrm{d}s]^2 \mathrm{d}x,$$

求 $y_N(x)$,使其满足

$$\min_{a_i^{(N)}} I(\boldsymbol{a}^{(N)}).$$

其中 $\boldsymbol{a}^{(N)} = (a_1^{(N)}, a_2^{(N)}, \cdots, a_N^{(N)})^{\mathrm{T}}$

根据式(14.4.1)$y_N(x)$的形式,置

$$\frac{\partial I}{\partial a_i^{(N)}} = 0, \quad i = 1,2,\cdots,N,$$

得到关于 $a_i^{(N)}$ 的线性方程组

$$\boldsymbol{L}_{\mathrm{LS}}^{(N)} \boldsymbol{a}^{(N)} = \boldsymbol{f}_{\mathrm{LS}}^{(N)}, \tag{14.4.2}$$

其中

$$(\boldsymbol{L}_{\mathrm{LS}}^{(N)})_{ij} = \int_a^b L h_i(x) \cdot L h_j(x) \mathrm{d}x, \quad i,j = 1,2,\cdots,$$

$$(\boldsymbol{f}_{\mathrm{LS}}^{(N)})_i = \int_a^b f(x) L h_i(x) \mathrm{d}x, \quad i = 1,2,\cdots,N,$$

$$Lh_i(x) = h_i(x) - \int_a^b K(x,s)h_i(s)\mathrm{d}s, \quad i = 1,2,\cdots,N.$$

方程组(14.4.2)是线性方程组，但形成矩阵 $L_{LS}^{(N)}$ 时，需要计算多重积分. 特别地，$L_{LS}^{(N)}$ 包含了如下的项，

$$\int_a^b \left[\int_a^b K(x,s)h_i(s)\mathrm{d}s\right] \cdot \left[\int_a^b K(x,s')h_j(s')\mathrm{d}s'\right]\mathrm{d}x.$$

这样的项有 N^2 个. 这样，积分近似计算关系到最小二乘法的工作量.

例 14.4.1 用最小二乘法计算积分方程

$$y(x) = \sin x - \frac{1}{4}x + \frac{1}{4}\int_0^{\frac{\pi}{2}} xsy(s)\mathrm{d}s$$

的近似解.

解 取 $h_1(x)=x, h_2(x)=x^3$，有

$$y(x) \approx y_2(x) = a_{x1}x + a_{x2}x^3$$

和

$$Lh_i(x) = h_i(x) - \frac{1}{4}\int_0^{\frac{\pi}{2}} xsh_i(s)\mathrm{d}s.$$

由此可得到

$$Lh_1(x) = x - \frac{1}{4}\int_0^{\frac{\pi}{2}} xs^2\mathrm{d}s = x\left(1 - \frac{\pi^3}{96}\right),$$

$$Lh_2(x) = x^3 - \frac{1}{4}\int_0^{\frac{\pi}{2}} xs^4\mathrm{d}s = x^3 - \frac{\pi^5 x}{5\times 128}.$$

这样可以得出线性方程组的系数矩阵及右端向量.

$$(\boldsymbol{L}_{LS})_{11} = \int_0^{\frac{\pi}{2}} (Lh_1(x))^2\mathrm{d}s \approx 0.59215955,$$

$$(\boldsymbol{L}_{LS})_{21} = (\boldsymbol{L}_{LS})_{12} = \int_0^{\frac{\pi}{2}} (Lh_1(x))(Lh_2(x))\mathrm{d}x \approx 0.87665709,$$

$$(\boldsymbol{L}_{LS})_{22} = \int_0^{\frac{\pi}{2}} (Lh_2(x))^2\mathrm{d}x \approx 1.83717689,$$

$$(\boldsymbol{f}_{LS})_1 = \int_0^{\frac{\pi}{2}} (Lh_1(x))\left(\sin x - \frac{1}{4}x\right)\mathrm{d}x \approx 0.45835330,$$

$$(f_{LS})_2 = \int_0^{\frac{\pi}{2}} (Lh_2(x))\left(\sin x - \frac{1}{4}x\right)\mathrm{d}x \approx 0.60032751.$$

把这些代入方程组(14.4.2),解出 a_{x1}, a_{x2},有

$$a_{x1} \approx 0.9887922, \quad a_{x2} \approx -0.1450618.$$

此例中积分方程的解析解为 $y(x)=\sin x$,取展开式的前两项有

$$\sin x \approx 1.0 - \frac{x^3}{6} = 1.0 - 0.166667x^3.$$

14.4.2 矩量法

用**矩量法**(method of moments)求积分方程(14.1.1)的近似解时,仍设 $\{h_i(x), i=1,2,\cdots\}$ 在 $\mathscr{L}^2(a,b)$ 中是完备的,并假定它们是正交的. 设 $\tilde{y}(x)$ 为方程(14.1.1)的近似解,那么

$$r(x) = f(x) - \left[\tilde{y}(x) - \lambda\int_a^b K(x,s)\,\tilde{y}(s)\mathrm{d}s\right]$$

称为 $\tilde{y}(x)$ 的**剩余函数**. 一般要求剩余函数为零. 由于 $\{h_i\}$ 在 $\mathscr{L}^2(a,b)$ 中完备、正交,因此 $r(x)=0$ 等价于 $r(x)$ 与 $h_i(x)(i=1,2,\cdots)$ 正交.

在求方程(14.4.1)的求近似解中,仅涉及到 N 个参数 $a_i^{(N)}(i=1,2,\cdots,N)$,因此最好使得

$$r_N(x) = f(x) - \left[y_N(x) - \lambda\int_a^b K(x,s)y_N(s)\mathrm{d}s\right]$$

与前 N 个函数 $h_1(x),\cdots,h_N(x)$ 正交,即

$$\int_a^b h_i(x)r_N(x)\mathrm{d}x = 0, \quad i=1,2,\cdots,N.$$

上面 N 个式子就导出了线性方程组

$$\boldsymbol{L}_G^{(N)}\boldsymbol{a}^{(N)} = \boldsymbol{f}_G^{(N)}, \tag{14.4.3}$$

其中

$$(\boldsymbol{L}_G^{(N)})_{ij} = \int_a^b h_i(x)h_j(x)\mathrm{d}x - \int_a^b\int_a^b h_i(x)K(x,s)h_j(s)\mathrm{d}x\mathrm{d}s,$$

$$(\boldsymbol{f}_G)_i = \int_a^b h_i(x) f(x) \mathrm{d}x, \quad i,j = 1,2,\cdots,N. \tag{14.4.4}$$

方程组(14.4.3)与用最小二乘法导出的方程组相似,但矩量法在结构上更简单.

一般函数系$\{h_i\}$是完备的,但不正交.此情况可以用格拉姆-施密特过程正交化.

假定$\{h_i\}$是线性无关的,$\{\bar{h}_i\}$是格拉姆-施密特过程形成的正交函数系.

$$\bar{h}_i(x) = \sum_{j=1}^{i} T_{ij} h_j, \tag{14.4.5}$$

其中$\boldsymbol{T}$是一个(三角形)正交化矩阵.

函数系$\{\bar{h}_i, i=1,2,\cdots,N\}$和$\{h_i, i=1,2,\cdots,N\}$张成同样的空间.令

$$\begin{aligned} y_N(x) &= \sum_{i=1}^{N} a_i h_i(x) = \sum_{i=1}^{N} \bar{a}_i \bar{h}_i(x) = \sum_{i=1}^{N} \bar{a}_i \sum_{j=1}^{i} T_{ij} h_j(x) \\ &= \sum_{j=1}^{N} \left(\sum_{i=j}^{N} \bar{a}_i T_{ij} \right) h_j(x), \end{aligned} \tag{14.4.6}$$

所以有

$$a_j = \sum_{i=j}^{N} T_{ij} \bar{a}_i,$$

即

$$\boldsymbol{a} = \boldsymbol{T}^{\mathrm{T}} \bar{\boldsymbol{a}}.$$

现在使剩余函数$r_N(x)$正交于$\{\bar{h}_i\}$中每一函数,如前面推导有

$$\bar{\boldsymbol{L}}_G^{(N)} \bar{\boldsymbol{a}}^{(N)} = \bar{\boldsymbol{f}}_G^{(N)},$$

其中

$$\begin{aligned} (\bar{\boldsymbol{L}}_G^{(N)})_{ij} &= \int_a^b \bar{h}_i(x) \left[\bar{h}_j(x) - \int_a^b K(x,s) \bar{h}_j(s) \mathrm{d}s \right] \mathrm{d}x \\ &= \sum_{k=1}^{i} \sum_{l=1}^{j} T_{ik} T_{jl} \int_a^b h_k(x) \left[h_l(x) - \int_a^b K(x,s) h_l(s) \mathrm{d}s \right] \mathrm{d}x \\ &= \sum_{k=1}^{i} \sum_{l=1}^{j} T_{ik} (\boldsymbol{L}_G^{(N)})_{kl}. \end{aligned}$$

由此可知，

$$\bar{L}_G^{(N)} = TL_G^{(N)}T^{\mathrm{T}},$$

$$\bar{f}_G^{(N)} = Tf_G^{(N)}.$$

最后得到：

$$TL_G^{(N)}a^{(N)} = Tf_G^{(N)}, \quad L_G^{(N)}a^{(N)} = f_G^{(N)}.$$

可以看出，此方程组是与前面一样的.

例 14.4.2 用矩量法求例 14.4.1 中的问题.

解 用例 14.4.1 中同样的近似解形式

$$y_G(x) = a_{G_1}x + a_{G_2}x^3,$$

其系数 a_{G_1}，a_{G_2} 用方程组(14.4.3)来求解. 函数

$$h_j(x) - \int_a^b K(x,s)h_j(s)\mathrm{d}s = Lh_j(x), \quad j = 1,2$$

已在例 14.4.1 中给出，把它们代入式(14.4.4)，有

$$(L_G)_{11} \approx 0.8746586, \quad (L_G)_{12} = (L_G)_{21} \approx 1.2948801,$$

$$(L_G)_{22} \approx 2.4563313;$$

$$f_{G_1} \approx 0.6770180, \qquad f_{G_2} \approx 0.9240475.$$

求解线性方程组(14.4.3)得

$$a_{G_1} \approx 0.9887920, \quad a_{G_2} \approx -0.1450620.$$

可以看出，其解与最小二乘近似得到的解相当符合.

14.5 特征值问题

用数值方法求解积分方程

$$\int_a^b K(x,s)y(s)\mathrm{d}s = Ky(x) \tag{14.5.1}$$

的特征值 κ 和特征函数. 其中 a，b 为有限的常数，$K(x,s)$ 为给定

的核.

求解方程(14.5.1)的数值方法将会产生有限个近似特征值 $\tilde{\kappa}_1,\tilde{\kappa}_2,\cdots,\tilde{\kappa}_n$ 以及相应的特征函数. 由于方程(14.5.1)将有可数的无限多个解,因此我们不能完全地求解. 通常可以得到模最大的特征值及相应的特征向量. 如果两个或多个特征值相互靠近,或有多重特征值,那么求相应的近似特征函数将很困难.

数值积分方法来求解方程(14.5.1)的近似特征值和近似特征向量并不是很好的方法,但此方法是最直接和最简单的.

选取适当的数值求积公式

$$\int_a^b F(x)\mathrm{d}x \approx \sum_{k=1}^n A_k F(x_k) \tag{14.5.2}$$

代入方程(14.5.1)中的积分,得

$$\sum_{k=1}^n A_k K(x,x_k)\,\tilde{y}(x_k) = \tilde{\kappa}\tilde{y}(x),$$

其中 $\tilde{\kappa}$, $\tilde{y}$ 分别是积分方程(14.5.1)中的 κ 和 y 的近似值. 特别地,令 $x=x_1,x_2,\cdots,x_n$,则得到一个代数特征值问题

$$\sum_{k=1}^n A_k K(x_j,x_k)\,\tilde{y}(x_k) = \tilde{\kappa}\tilde{y}(x_j) \quad (j=1,2,\cdots,n). \tag{14.5.3}$$

此式可以写成矩阵形式

$$\boldsymbol{KD}\,\tilde{\boldsymbol{y}} = \tilde{\kappa}\,\tilde{\boldsymbol{y}}, \tag{14.5.4}$$

其中

$$\boldsymbol{K} = \begin{pmatrix} K(x_1,x_1) & K(x_1,x_2) & \cdots & K(x_1,x_n) \\ K(x_2,x_1) & K(x_2,x_2) & \cdots & K(x_2,x_n) \\ \vdots & \vdots & & \vdots \\ K(x_n,x_1) & K(x_n,x_2) & \cdots & K(x_n,x_n) \end{pmatrix},$$

$$\boldsymbol{D} = \mathrm{diag}(A_1,A_2,\cdots,A_n),$$

$$\tilde{\boldsymbol{y}} = (\tilde{y}_1, \tilde{y}_2, \cdots, \tilde{y}_n)^{\mathrm{T}}, \quad \tilde{y}_j = \tilde{y}(x_j).$$

如果核 $\boldsymbol{K}$ 是实对称的，那么可以把式(14.5.4)改写为

$$(\boldsymbol{D}^{\frac{1}{2}}\boldsymbol{K}\boldsymbol{D}^{\frac{1}{2}})\,\tilde{\boldsymbol{z}} = \tilde{\kappa}\tilde{\boldsymbol{z}}, \qquad (14.5.5)$$

其中$\tilde{\boldsymbol{z}} = \boldsymbol{D}^{\frac{1}{2}}\tilde{\boldsymbol{y}}$. 由于矩阵 $\boldsymbol{D}^{\frac{1}{2}}\boldsymbol{K}\boldsymbol{D}^{\frac{1}{2}}$ 是实对称矩阵，因此特征向量和特征值容易求出.

例 14.5.1 求积分方程

$$\int_0^1 (1 - 3xs)\,y(s)\,\mathrm{d}s = \kappa y(x)$$

的近似特征值和相应的近似特征函数.

解 取求积公式(14.5.2)为梯形公式，并把它代入积分方程，那么对每个 x 得

$$\frac{1}{2}\,\tilde{y}(0) + \frac{1}{2}(1 - 3x)\,\tilde{y}(1) = \tilde{\kappa}\,\tilde{y}(x). \qquad (14.5.6)$$

特别地，取 $x=0,1$，则得方程组

$$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -1 \end{bmatrix} \begin{bmatrix} \tilde{y}(0) \\ \tilde{y}(1) \end{bmatrix} = \tilde{\kappa} \begin{bmatrix} \tilde{y}(0) \\ \tilde{y}(1) \end{bmatrix}.$$

容易求出特征值为$-\frac{1}{4} \pm \frac{\sqrt{13}}{4}$，相应的特征函数$\tilde{y}$可由$\tilde{y}(0)$，$\tilde{y}(1)$及式(14.5.6)得到.

由于梯形公式误差较大，因此所得到的特征值及相应的特征函数误差也较大.

取 $h=\frac{1}{2}$的复化梯形公式，可以得到

$$\frac{1}{4}\,\tilde{y}(0) + \frac{1}{2}\left(1 - \frac{3}{2}x\right)\tilde{y}\left(\frac{1}{2}\right) + \frac{1}{4}(1 - 3x)\,\tilde{y}(1) = \tilde{\kappa}\,\tilde{y}(x).$$

取 $x=0,\frac{1}{2},1$，则得线性代数方程组

$$\begin{pmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{8} & -\frac{1}{8} \\ -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{2} \end{pmatrix} \begin{pmatrix} \tilde{y}(0) \\ \tilde{y}\left(\frac{1}{2}\right) \\ \tilde{y}(1) \end{pmatrix} = \tilde{\kappa} \begin{pmatrix} \tilde{y}(0) \\ \tilde{y}\left(\frac{1}{2}\right) \\ \tilde{y}(1) \end{pmatrix}$$

或可以写做对称形式

$$\begin{pmatrix} \frac{1}{4} & \frac{\sqrt{2}}{4} & \frac{1}{4} \\ \frac{\sqrt{2}}{4} & \frac{1}{8} & -\frac{\sqrt{2}}{8} \\ \frac{1}{4} & -\frac{\sqrt{2}}{8} & -\frac{1}{2} \end{pmatrix} \begin{pmatrix} \tilde{z}(0) \\ \tilde{z}\left(\frac{1}{2}\right) \\ \tilde{z}(1) \end{pmatrix} = \tilde{\kappa} \begin{pmatrix} \tilde{z}(0) \\ \tilde{z}\left(\frac{1}{2}\right) \\ \tilde{z}(1) \end{pmatrix},$$

其中

$$\left(\tilde{z}(0), \tilde{z}\left(\frac{1}{2}\right), \tilde{z}(1)\right)^{\mathrm{T}} = \left(\frac{1}{2}\tilde{y}(0), \frac{1}{\sqrt{2}}\tilde{y}\left(\frac{1}{2}\right), \frac{1}{2}\tilde{y}(1)\right)^{\mathrm{T}}.$$

可以求出其特征值为 $-\frac{1}{16} \pm \frac{\sqrt{26}}{8}, 0$，仿照前面方法可以求出相应的特征函数.

14.6 第二类沃尔泰拉积分方程的数值积分法

考虑第二类**线性沃尔泰拉积分方程**(linear Volterra integral equation of second kind)

$$y(x) = \int_a^x K(x,s)y(s)\mathrm{d}s + f(x), \qquad (14.6.1)$$

假定其解是要求在有限区间 $[a,b]$ 上，$f(x)$ 在 $[a,b]$ 上连续，$K(x,s)$ 在 $a \leqslant s \leqslant x \leqslant b$ 上连续.

14.6.1 用多步法解第二类沃尔泰拉积分方程

设 $x=nh$，采用等距节点 $x_0=0, x_k=kh, k=1,2,\cdots,n$. 相应求积公式记为

$$\int_0^x F(s)\mathrm{d}s \approx \sum_{k=0}^{n} A_{nk}F(x_k) = \sum_{k=0}^{n} A_{nk}F(kh). \quad (14.6.2)$$

把公式(14.6.2)代人积分方程(14.6.1)的积分中，可以得到

$$\tilde{y}(nh) = \sum_{k=0}^{n} A_{nk}K(nh,kh)\tilde{y}(kh) + f(nh).$$

把上式改写为多步法的一般公式，有

$$\begin{aligned}&[1-A_{nn}K(nh,nh)]\tilde{y}(nh)\\ &= \sum_{k=0}^{n-1} A_{nk}K(nh,kh)\tilde{y}(kh) + f(nh). \quad (14.6.3)\end{aligned}$$

特别地，把等距求积公式(14.6.2)取为复化梯形公式，那么，公式(14.6.3)简化为

$$\left[1-\frac{1}{2}hK(nh,nh)\right]\tilde{y}(nh) = h\sum_{k=1}^{n-1}K(nh,kh)\tilde{y}(kh) + \frac{1}{2}hK(nh,0)\tilde{y}(0) + f(nh). \quad (14.6.4)$$

由于从积分方程可以得到 $\tilde{y}(0)=f(0)$，因此这个公式开始计算特别容易，$\tilde{y}(h), \tilde{y}(2h), \cdots, \tilde{y}(nh)$ 都可从公式(14.6.4)递推地求出.

例 14.6.1 在区间[0,2]上，使用 $n=20$ 的复化梯形公式求积分方程

$$y(x) = \int_0^x \frac{1}{5}xsy(s)\mathrm{d}s + x, \quad x>0$$

的近似解.

解 首先,由于$\tilde{y}(0)=0$,利用公式(14.6.4)有

$$\left[1-0.05\times\frac{1}{5}(0.1)^2\right]\tilde{y}(0.1)$$
$$=0.1+0.05\times\frac{1}{5}\times 0.1\times 0\times\tilde{y}(0)=0.1,$$

因此

$$\tilde{y}(0.1)=0.10001.$$

同样可以得出

$$\tilde{y}(0.2)=0.20012,\quad \tilde{y}(0.3)=0.30057,\cdots,$$
$$\tilde{y}(1.9)=3.00564,\quad \tilde{y}(2)=3.41463.$$

此例积分方程的精确解为

$$y(x)=x\exp\left(\frac{x^3}{15}\right).$$

因此可以得到误差估计

$$\tilde{y}(0.3)-y(0.3)=0.30057-0.30054=0.00003,$$
$$\tilde{y}(1.9)-y(1.9)=3.00564-3.00152=0.00412.$$

为了更精确地计算,必须用高阶求积公式来代替复化梯形公式,但是,使用高阶求积公式必须有一个特殊的开始方法.最常用的是**戴依方法**(Day method),计算$\tilde{y}(h)$,$\tilde{y}(2h)$,$\tilde{y}(3h)$的公式有:

$$\tilde{y}(h)=\frac{1}{6}h\Big[K(h,0)f(0)+4K\left(h,\frac{1}{2}h\right)y_{13}+K(h,h)y_{12}\Big]+f(h),\tag{14.6.5}$$

其中

$$y_{11}=hK(h,0)f(0)+f(h),$$
$$y_{12}=\frac{1}{2}h[K(h,0)f(0)+K(h,h)y_{11}]+f(h),$$
$$y_{13}=\frac{1}{4}h\Big[K\left(\frac{1}{2}h,0\right)f(0)+$$

$$K\left(\frac{1}{2}h,\frac{1}{2}h\right)\left(\frac{1}{2}f(0)+\frac{1}{2}y_{12}\right)\Big]+f\left(\frac{1}{2}h\right),$$

$$\tilde{y}(2h)=\frac{1}{3}h\Big[K(2h,0)f(0)+4K(2h,h)\,\tilde{y}(h)+K(2h,2h)y_{21}\Big]+f(2h), \tag{14.6.6}$$

其中

$$y_{21}=2hK(2h,h)\,\tilde{y}(h)+f(2h),$$

$$\tilde{y}(3h)=\frac{3}{8}h\Big[K(3h,0)f(0)+3K(3h,h)\,\tilde{y}(h)+3K(3h,2h)\,\tilde{y}(2h)+K(3h,3h)y_{31}\Big]+f(3h), \tag{14.6.7}$$

其中

$$y_{31}=\frac{3}{2}h[K(3h,h)\,\tilde{y}(h)+K(3h,2h)\,\tilde{y}(2h)]+f(3h).$$

戴依方法与复化辛普森求积公式的联合使用,改善了求解的精度.但是,由于复化辛普森求积公式仅适用于奇数个等距节点,所以,对偶数个节点的情况,可采用 3/8 辛普森求积公式.

例 14.6.2 在区间[0,2]上使用节点距 $h=0.1$ 的辛普森求积公式和 3/8 辛普森求积公式计算积分方程

$$y(x)=\int_0^x \frac{1}{5}xsy(s)\mathrm{d}s+x,\quad x>0$$

的近似解.

解 由于 $h=0.1$, $K(x,s)=\frac{1}{5}xs$,所以首先采用戴依方法来确定 $\tilde{y}(h)$,得

$$y_{11}=0.1,$$

$$y_{12}=0.1+0.05\left[0+\frac{1}{5}(0.1)^3\right]=0.10001,$$

$$y_{13}=0.05+0.025\left[0+\frac{1}{5}(0.05)^2\times(0+0.05)\right]\approx 0.05,$$

$$\tilde{y}(h)=\tilde{y}(0.1)$$

$$=0.1+\frac{1}{6}(0.1)\left[0+4\times\frac{1}{5}\times0.1\times0.05\times0.05+\frac{1}{5}\times0.1\times0.1\times0.10001\right]$$

$$=0.10001.$$

利用辛普森公式计算，有

$$\tilde{y}(2h)=\tilde{y}(0.2)=0.20011.$$

再用 3/8 辛普森公式计算，有

$$\tilde{y}(3h)=\tilde{y}(0.3)=0.30054.$$

在整个求解过程中，$\tilde{y}(0.4),\tilde{y}(0.6),\cdots,\tilde{y}(2.0)$是用辛普森公式计算，而$\tilde{y}(0.5),\tilde{y}(0.7),\cdots,\tilde{y}(1.9)$则皆用 3/8 辛普森公式进行计算.

为进行比较，在表 14.1 中列出了本例的数值结果、积分方程的精确解以及用复化梯形求积得到的结果.

表 14.1

x_k	精确解	梯　形	辛普森
0.0	0.00000	0.00000	0.00000
0.1	0.10001	0.10001	0.10001
0.2	0.20011	0.20012	0.20011
0.3	C.30054	0.30057	0.30054
0.6	0.60870	0.60883	0.60870
0.9	0.94482	0.94514	0.94482
1.2	1.34652	1.34720	1.34652
1.5	1.87849	1.87993	1.87849
1.8	2.65538	2.65582	2.65538
1.9	3.00152	3.00564	3.00154
2.0	3.40921	3.41463	3.40922

表 14.1 虽然没有把全部计算结果列出，但可以看出，采用辛普森公式比复化梯形公式要精确得多.

$$y_n^{(4)} = F_n(x_{n+1}) + \frac{1}{6}h\{K(x_{n+1}, x_n)y_n^{(0)} + 2K(x_{n+1}, x_{n+\frac{1}{2}})y_n^{(1)} + 2K(x_{n+1}, x_{n+\frac{1}{2}})y_n^{(2)} + K(x_{n+1}, x_{n+1})y_n^{(3)}\}, \tag{14.6.9}$$

那么,第二类沃尔泰拉积分方程在 $x = x_{n+1}$ 的近似解 $\tilde{y}$ 可以表示为

$$\tilde{y}(x_{n+1}) = y_n^{(4)}. \tag{14.6.10}$$

例 14.6.3 用龙格-库塔型方法求积分方程

$$y(x) = \int_0^x \frac{1}{5}xsy(s)\mathrm{d}s + x, \quad x > 0$$

的解在 $x = x_k = kh(h = 0.1, k = 0, 1, 2, \cdots, 20)$ 的近似值.

解 按公式计算,得

$$y_0^{(0)} = 0,$$

$$y_0^{(1)} = 0.05 + 0.05 \times \frac{1}{5} \times 0.05 \times 0 \times 0 = 0.05,$$

$$y_0^{(2)} = 0.05 + 0.05 \times \frac{1}{5} \times 0.05^3 \approx 0.05,$$

$$y_0^{(3)} = 0.1 + 0.1 \times \frac{1}{5} \times 0.1 \times 0.05^2 \approx 0.10001,$$

于是有

$$\tilde{y}(0.1) \approx y_0^{(4)} = 0.1 + \frac{1}{6} \times 0.1 \times (\frac{1}{5} \times 0.1 \times 0 \times 0 + 2 \times \frac{1}{5} \times 0.1 \times 0.05 \times 0.05 + 2 \times \frac{1}{5} \times 0.1 \times 0.05 \times 0.05 + \frac{1}{5} \times 0.1 \times 0.1 \times 0.10001) = 0.10001.$$

类似地,

$$y_1^{(0)} = 0.10001, \quad y_1^{(1)} = 0.15002,$$
$$y_1^{(2)} = 0.15004, \quad y_1^{(3)} = 0.20010,$$

于是有

$$\tilde{y}(0.2) \approx y_1^{(4)} = 0.20010.$$

14.6.2 用龙格-库塔型方法解第二类沃尔泰拉积分方程

积分方程

$$y(x)=\int_0^x (s-x)y(s)\mathrm{d}s+\alpha+\beta x$$

等价于常微分方程的初值问题

$$y''(x)+y(x)=0,\quad y(0)=\alpha,\quad y'(0)=\beta.$$

由此可以采用类似于求解常微分方程的办法来求解积分方程. 采用高阶求积公式的多步法必须使用特别的开始步骤,这是很不方便的. 此外,改变步长也不容易. 采用龙格-库塔型方法来求解积分方程

$$y(x)=\int_0^x k(x,s)y(s)\mathrm{d}s+f(x).$$

可以克服多步法的缺点,用龙格-库塔型方法来求解,有不同的形式. 例如,使用**布泽公式**(Pouzet formula). 令

$$\begin{cases} y_n^{(0)}=y_{n-1}^{(4)}\quad (y_0^{(0)}=f(x_0)),\\ y_n^{(1)}=F_n(x_{n+\frac{1}{2}})+\dfrac{1}{2}hK(x_{n+\frac{1}{2}},x_n)y_n^{(0)},\\ y_n^{(2)}=F_n(x_{n+\frac{1}{2}})+\dfrac{1}{2}hK(x_{n+\frac{1}{2}},x_{n+\frac{1}{2}})y_n^{(1)},\\ y_n^{(3)}=F_n(x_{n+1})+hK(x_{n+1},x_{n+\frac{1}{2}})y_n^{(2)}, \end{cases}\tag{14.6.8}$$

其中

$$\begin{aligned} F_n(x)=&f(x)+\frac{1}{6}h\sum_{j=0}^{n-1}\{K(x,x_j)y_j^{(0)}+\\ &2K(x,x_{j+\frac{1}{2}})y_j^{(1)}+2K(x,x_{j+\frac{1}{2}})y_j^{(2)}+\\ &K(x,x_{j+1})y_j^{(3)}\},\\ F_0(x)=&f(x). \end{aligned}$$

设

其余近似值仿此可求出.在一些点上计算的结果见表 14.2.

表 14.2

x_k	0.1	0.4	0.7	1.0
计算解	0.10001	0.40171	0.71619	1.06894
精确解	0.10001	0.40171	0.71619	1.06894
x_k	1.3	1.6	1.9	2.0
计算解	1.50506	2.10238	3.00152	3.40920
精确解	1.50506	2.10238	3.00152	3.40921

由数值结果可以看出,这个方法相当精确,但是在计算过程中,函数值计算次数多,耗时也多.



附录　数学软件 Matlab 简介

Matlab 是 Matrix Laboratory(矩阵实验室)的缩写,它是由美国 MathWork 公司于 1967 年推出的数学软件,现已发展成为适用于多种类型计算机的软件平台,而且是集数值计算功能、符号计算功能和计算机可视化为一体的最强大的、最有吸引力和最广泛的科学计算语言.这里仅给出与本手册内容有关的函数命令及注释,使用时需深入了解每条命令的输入、输出参数或格式,可借助于 Matlab 软件的"help"功能或参考有关资料.

1　矩阵和矩阵分析

zeros	零矩阵
ones	全 1 矩阵
eye	单位矩阵
hilb	希尔伯特矩阵
toeplits	特普利茨矩阵
vander	范德蒙矩阵
jordan	若尔当标准型
rand	均匀分布随机阵
randn	正态分布随机阵
size	矩阵大小
length	矩阵长度
ndims	维数
diag	生成对角阵和矩阵对角向量
tril	产生(提取)下三角部分
triu	产生(提取)上三角部分

rank	矩阵的秩
trace	矩阵的迹
norm	矩阵或向量范数
cond	矩阵的条件数
condest	矩阵的 1-条件数
rcond	条件数的倒数
det	矩阵行列式的值
inv	非奇异方阵的逆矩阵

2 插值和拟合

interp1	一维插值
interp2	二维插值
interp3	二维双调和数据插值
interp4	二维双线性插值
interp5	二维双三次插值
interft	一维 FFT 插值
icubic	一维三次插值
spline	三次样条插值
ppval	分段多项式插值
polyfit	多项式曲线拟合
leastsq	非线性最小二乘曲线拟合

3 数值积分

quad	自适应辛普森法
quad8	自适应牛顿-科茨法
dblquad	二重积分

4 方程求根

roots	求多项式的根(实和复)
fzero	单变量函数的根
fmin	单变量函数的局部极小值

fmins	多变量函数的局部极小值

5　线性方程组的直接法

A\b,b/A	列主元高斯消去法求方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 的解向量 $\boldsymbol{x}$
nnls	在最小二乘意义下求方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 的解
lu	高斯消去法对矩阵作 LU 分解
luinc	矩阵的不完全 LU 分解
chol	对称正定矩阵的楚列斯基分解
cholinc	对称正定矩阵的不完全楚列斯基分解
qr	矩阵的 QR 分解

6　线性方程组的迭代法

pcg	预处理共轭梯度法
bicg	B_i 共轭梯度法
bicgstab	B_i 稳定共轭梯度法
cgs	二次共轭梯度法
gmres	广义极小残余法即 GMRES 法
gmr	准最小残余量

7　矩阵的特征值和特征向量计算

poly	特征多项式
eig	特征值和特征向量
eigs	特征值
eig(A,B)	$(\boldsymbol{A},\boldsymbol{B})$的广义特征值
[u,l]=eig(A,B)	矩阵 $\boldsymbol{u}$ 的列为$(\boldsymbol{A},\boldsymbol{B})$的广义特征向量,$l$ 为广义特征值
svd	奇异值分解
svds	奇异值
hess	对矩阵 $\boldsymbol{A}$,生成一个海森伯格形式的矩阵 $\boldsymbol{H}$ 及一个酉阵 $\boldsymbol{P}$,使 $\boldsymbol{A}=\boldsymbol{PHP}$. $\boldsymbol{A}$ 和 $\boldsymbol{H}$ 有

	相同的特征值
schur	Schur 分解. 产生一个 Schur 形矩阵 $\boldsymbol{T}$ 和一个酉阵 $\boldsymbol{U}$, 使 $\boldsymbol{A}=\boldsymbol{UTU}^{\mathrm{T}}$. 若 $\boldsymbol{A}$ 是复的, 得到一个复的 Schur 上三角阵 $\boldsymbol{T}$, 其对角线元素为 $\boldsymbol{A}$ 的特征值. 若 $\boldsymbol{A}$ 是实的, 得到一个实的 Schur 形矩阵, 对角线为 1 阶和 2 阶块矩阵

8　非线性方程组的数值解法

fsolve	非线性方程组数值解
numjac	数值计算雅可比矩阵

9　常微分方程数值解

ode23	用 2 阶和 3 阶龙格-库塔法求解非刚性常微分方程初值问题
ode23s	同 ode23 且给出结果的图形
ode45	用 4 阶和 5 阶龙格-库塔法求解非刚性常微分方程初值问题
ode113	用变阶方法求解非刚性常微分方程初值问题
ode15s	同 ode113 且给出结果的图形
bvp4c	用配置法求解常微分方程边值问题
odephas2	二维相平面输出函数
odephas3	三维输出函数

注: 如果没有判定问题是否为刚性的, 则首先试用 ode45, 然后再试用 ode15s.

10　偏微分方程数值解

poisolv	二维泊松方程的数值解

详见 Matlab PDE 工具箱

11 其他软件包和软件

Mathematic	长于符号计算
Maple	长于符号计算
Linpack	线性代数方程组的数值解
Eispack	数值计算稠密特征值问题
LApack	基于 Linpack 和 Eispack 且吸收了更多现代算法. 达到更高精度,更健壮,有更多功能
Netlib	有计算大型稀疏特征值问题的兰乔斯算法软件
Arpack	有计算大型稀疏非对称特征值问题的 Arnoldi 算法软件
Svdpack	有计算大型稀疏矩阵的奇异值和向量的兰乔斯法和子空间迭代法
Pcgpack	线性方程组的预处理共轭梯度法
ELLpack	求解各种二维和三维椭圆型边值问题的专用软件包
Fishpack	在各种坐标下快速求解二维亥姆霍兹解算器



外国人名表

A	Adams, J. C.	亚当斯
	Aitken, A. C.	艾特肯
	Arnoldi	阿诺尔德
B	Bairstow, L.	贝尔斯托
	Bashforth, F.	巴什福思
	Bauer, F. L.	鲍尔
	Beam	比姆
	Bernoulli, J.	伯努利
	Bessel, F. W.	贝塞尔
	Brent, P. R.	布伦特
	Brown, G. W.	布朗
	Brouwer	布劳维尔
	Broyden, C. G.	布罗依登
	Bui, T. D.	布意
	Butcher, J. C.	布特切尔
C	Cauchy, A. L.	柯西
	Chebyshev/Чебышев	切比雪夫
	Cholesky	楚列斯基
	Cohen, C. J.	科恩
	Concus, P.	康卡斯
	Cotes, R.	科茨
	Courant, R.	柯朗
	Cramer, G.	克拉默
	Crank, J.	克拉克
	Crout, P. D.	克劳特
D	D'Alembert, J. L. R.	达朗贝尔
	Davidon, W. C.	达维顿
	Day, M. M.	戴
	Dirichlet, P. G. L.	狄利克雷

	Doolittle	杜利特尔
	Douglas,J	道格拉斯
	DuFort,E. C.	杜福特
E	Eaves,B.	伊维斯
	Euclid	欧几里得
	Euler,L.	欧拉
	Everett,J. D.	埃弗雷特
F	Fehlberg,E.	费尔贝格
	Fike,C. J.	费凯
	Filon,L. N. G.	菲隆
	Fletcher,R.	弗莱彻
	Fourier,J. B. J.	傅里叶
	Frankel,S. P.	弗兰克尔
	Fraser	弗雷泽
	Fréchet,M.	弗雷歇
	Fredholm,E. I.	弗雷德霍姆
	Friedrichs,K. O.	弗里德里希斯
	Frobenius	弗罗贝尼乌斯
G	Galerkin	伽辽金
	Gargantini	伽甘蒂尼
	Gauss,K. F.	高斯
	Gâteaux,R.	加托
	Gear,C. W.	吉尔
	Gerschgorin,S.	格尔施戈林
	Gill,S.	基尔
	Givens,J.	吉文斯
	Goldfarb,D.	哥德法布
	Golub,G. H.	哥拉布
	Gragg,W. B.	格拉格
	Gram,J. P.	格拉姆
	Gregory,J.	格雷戈里
H	Haar	哈尔
	Hamming,R. W.	汉明
	Hammer,P. C.	哈默
	Henrici,P.	亨里奇
	Hermite,C.	埃尔米特

	Hesse, H.	黑塞
	Hessenberg, K.	海森伯格
	Heun, K.	休恩
	Hilbert, D.	希尔伯特
	Hollingsworth	荷令斯沃思
	Hölder, O.	赫尔德
	Horner, W. G.	霍纳
	Householder, A. S.	豪斯霍尔德
	Huta, A.	休塔
J	Jacobi, C. G. J.	雅可比
	Johnsson, S. L.	约翰逊
	Jordan, C.	若尔当
K	Kontorovich	康托洛维奇
	Krawczyk, R.	克拉夫楚克
	Kroneker, L.	克罗内克
	Krylov	克雷洛夫
	Kuhn, H.	库恩
	Kutta, W.	库塔
L	Lagrange, J. I.	拉格朗日
	Laguerre, E. N.	拉盖尔
	Lanczos, C.	兰乔斯
	Laplace, P. S. M.	拉普拉斯
	Lax, P. D.	拉克斯
	Legendre, A. M.	勒让德
	Lewy, H.	勒维
	Liebman, K. O. H.	李普曼
	Lipschitz, R. O. S.	利普希茨
	Lobatto	洛巴托
M	Maclaurin, C.	麦克劳林
	Maehly	梅利
	Markov, A. A.	马尔可夫
	Merrill, O.	默里尔
	Milne, W. E.	米尔恩
	Moore, R.	穆尔
	Morison, D.	莫里森
	Moulton, F. R.	莫尔顿

	Müller, D. E.	米勒
N	Newton, I.	牛顿
	Nickel, K.	尼克尔
	Nicolson, P.	尼科尔森
	Nyström, E. J.	尼斯特龙
P	Padé, H.	帕德
	Patterson, L. N.	帕特森
	Peaceman, D. W.	皮斯曼
	Picard	皮卡
	Poisson, S. D.	泊松
	Pouzet	布泽
	Powell, M. B.	鲍威尔
R	Rachford, H. H.	瑞奇福尔德
	Radau, R.	拉道
	Ralston, A.	赖尔斯顿
	Rayleigh, J. W. S.	瑞利
	Remes, E. J.	列梅兹
	Riccati, J. F.	里卡蒂
	Richardson, L. F.	里查森
	Riemann, G. F. B.	黎曼
	Ritz, W.	里茨
	Roberts, K. V.	罗伯茨
	Robinson	鲁宾逊
	Romberg, W.	龙贝格
	Rolle, M.	罗尔
	Runge, C.	龙格
	Ruth	鲁思
S	Samarskii	萨马尔斯基
	Scarf, H.	斯卡夫
	Schmidt	施密特
	Schur	舒尔
	Schwartz, H. A.	施瓦兹
	Seidel, P. L.	赛德尔
	Shannon, D. F.	香农
	Sherman, A.	谢尔曼
	Simpson, T.	辛普森

	Sperner, E.	斯珀内
	Steffensen, J. F.	斯蒂芬森
	Stirling, J.	斯特林
	Sturm, J. C. F.	施图姆
T	Taylor, B.	泰勒
	Thiele, T. N.	蒂埃勒
	Thomas, J. W.	托马斯
V	Volterra, V.	沃尔泰拉
	von Neumann, J.	冯·诺伊曼
W	Warming	沃明
	Weierstrass, K.	魏尔斯特拉斯
	Weilandt	维兰特
	Weiss, N. O.	维斯
	Wendroff, B.	温德洛夫
	Wilkinson, J.	威尔金森
	Widlund, O.	怀德伦德
	Woodburg	伍德伯格
	Бернштейн, С. Н.	伯恩斯坦
	Мысовский, И. П.	梅索夫斯基
	Шаманский, В.	沙曼斯基
	Яненко	扬年科

外文—中文名词索引

A

B

C

D

E

J

K

L

N

Q

R

S

T

U

V

W

Z

中文—外文名词索引

A

B

C

D

F

G

H

J

K

L

M

N

O

P

Q

T

R

S

U

W

X

Y

Z

参考文献

0 清华大学应用数学系《现代应用数学手册》编委会. 现代应用数学手册——计算方法分册. 北京：北京出版社，1990

1 李庆扬，王能超，易大义编. 数值分析. 第4版、北京：清华大学出版社，2001

2 李庆扬，关治，白峰杉编著. 数值计算原理. 北京：清华大学出版社，2000

3 关治，陆金甫. 数值分析基础. 北京：高等教育出版社，1998

4 关治，陈景良. 数值计算方法. 北京：清华大学出版社，1990

5 冯康等编. 数值计算方法. 北京：国防工业出版社，1978年

6 施妙根，顾丽珍. 科学和工程计算基础. 北京：清华大学出版社，1999

7 蔡大用，白峰杉. 高等数值分析. 北京：清华大学出版社，1997

8 李庆扬，易大义，王能超编. 现代数值分析. 北京：高等教育出版社，1995

9 清华大学、北京大学《计算方法》编写组编. 计算方法(上、下册). 北京：科学出版社，1974，1980

10 蔡大用，白峰杉. 现代科学计算. 北京：科学出版社.2000

11 蔡大用编著. 数值代数. 北京：清华大学出版社，1987

12 曹志浩编著. 数值线性代数. 上海：复旦大学出版社，1995

13 胡家赣著. 线性代数方程组的迭代方法. 北京：科学出版社，1991

14 冯果忱，于庚蒲，郃继福. 矩阵迭代分析导论. 长春：吉林大学出版社，1991

15 曹志浩，张玉德，李瑞瑕. 矩阵计算和方程求根. 北京：人民教育出版社，1979

16 冯果忱. 非线性方程组迭代解法. 上海：上海科学技术出版社，1989

17 王德人. 非线性方程组解法和最优化方法. 北京：人民教育出版社，1979

18 王仁宏著. 数值有理逼近. 上海：上海科学技术出版社，1980

19 李岳生，黄友谦. 数值逼近. 北京：人民教育出版社，1978

20 徐利治，王仁宏，周蕴时. 函数逼近的理论和方法. 上海：上海科技出版社，1983

21 李庆扬等.非线性方程组数值解法. 北京：科学出版社，1987
22 陆金甫，关治. 偏微分方程数值解. 第2版. 北京：清华大学出版社，2004
23 陆金甫，顾丽珍，陈景良编著. 偏微分方程差分方法. 北京：高等教育出版社，1988
24 李荣华，冯果忱. 偏微分方程数值解法. 北京：人民教育出版社，1980
25 姜礼尚，庞之恒. 有限元方法及其理论基础. 北京：人民教育出版社，1979
26 李庆扬. 常微分方程数值解法(刚性问题与边值问题). 北京：高等教育出版社，1991
27 Henrici P 著，包雪松等译. 常微分方程的离散变量方法. 北京：科学出版社，1985
28 Gear C W 著，费景高等译. 常微分方程初值问题的数值方法. 北京：科学出版社，1978
29 南京大学数学系. 常微分方程数值解法. 北京：科学出版社，1979
30 北京大学，吉林大学，南京大学编. 计算方法. 北京：高等教育出版社，1961
31 [美]G. H. 戈卢布，C. F. 范洛恩著，袁亚湘等译. 矩阵计算. 北京：科学出版社，2001
32 [德]Stoer J and Bulrisch R 著，孙文瑜等译. 数值分析引论. 南京：南京大学出版社，1995
33 Saad Y. Iterative Methods for Sparse Linear Systems. Philadelphia: Society for Industrial & Applied Mathematics，2nd ed，2003
34 Saad Y. Numerical Methods for large Eigenvalue Problem. Manchester university Press，1992
35 J. H. 威尔金森著，石钟慈，邓健新译. 代数特征值问题. 北京：科学出版社，1987
36 Varga R. S. Matrix Iterative Analysis. Prentice-Hall, Inc，Englewood Cliffs，New Jersey，1962
37 Michael T. Heath. Scientific Computing-An Introductory Survey. Second Edition(科学计算导论(第2版)). 北京：清华大学出版社，麦格劳-希尔教育出版社，2001，1997

38 Pissanetzky S. Sparse Matrix Technology. New York: Academic Press, 1984

39 [美]R. P. 梯华森著,朱季纳译. 稀疏矩阵. 北京：科学出版社,1981

40 《现代数学手册》编纂委员会. 现代数学手册——计算机数学卷. 武汉：华中科技大学出版社,2001

41 O'Leary D P and R E White. Multispliting of Matrices and Parallel of Linear Systems. SIAM J. Alg. Disc. Methods,1985,6:630～640

42 Frommer A. Parallel Nonlinear Multisplitting methods. Numer. Math., 1989, 56: 269～282

43 Ortega J M and Rheinboldt W C. Iterative Solution of Nonlinear Equations in Several Variable. New York: Academic Press, 1970

44 Wilkinson J. Rounding Error in Algebraic Processes, Notes on Applied Science No. 32. Her Majesty's Stationary Office, London: Prentice-Hall, New Jersey: Inc, Englewood Cliffs, 1963

45 Brenner S C and Scott L R. The Mathematical Theory of Finite Element Methods. New York: Springer-Verlag, 1994

46 Burden R L and Faires J D. Numerical Analysis. 7th ed.. Beijing: Thomson Learning and HEP, 2002

47 Ciarlet P G. The Finite Element for Eilliptic Problems. Amsterdam, New York, Oxford, North-Holland, 1978

48 Ciarlet P G and Lions J L. Handbook of Numerical Analysis, vol. Ⅱ. Amsterdam, New York,Oxford, North-Holland, 1991

49 Keller H B. Numerical Methods for Two-Point Boundary-value Problems. Waltham, Blaisdell, 1968

50 Lambert J D. Computional Methods in Ordinary Differential Equation. New York: John Wiley & sons, 1973

51 Noye J. Computional Techniques for Differential Equations. Amsterdam, New York, Oxford, North-Holland, 1984

52 Stoer J and Bulirsch R. Introduction to Numerical Analysis. 2nd ed.. New York: Springer-Verlag, 1993

53 Thomas J W. Numerical Partial Differential Equation. Heidelberg: Springer-Verlag, 1995

54 Delves L M and Mohamed J C. Computional Methods for Integral Equations. Cambridge: Cambridge University Press, 1985
55 Davis P J and Rabinowitz P. Methods of Numerical Integration. New York: Academic Press, 1975
56 Christopher T H and Baker M A. The Numerical Treatment of Integral Equation. Oxford: Clarendon Press, 1977
57 Leveque R J. Numerical Methods for Conservation Laws. Basel: Birkhäuser Verlag, 1990
58 Davis P J and Rabinowitz P. Methods of Numerical Integration. 2nd ed. New York: Academic Press, 1984
59 Brandt A. Multi-level adaptive Technique of fast numerical Solution to Boundary Value Problems. Proceeding of Third international Conference on Numerical Methods in Fluid Mechanics. Paris, 1972
60 Brandt A. Multi-level adaptive Solution to Boundary-Value Problem. Mathematics of Computation, 1977, 31(138): 333～390
61 Briggs W L A. Multigrid tutorial. Philadelphia: SIAM, 1987
62 Hackbusch W. Multigrid Method and Application. Berlin: Springer, 1985
63 W. 哈克布思著. 多重网格方法. 北京：科学出版社，1988
64 McCormick S F. Multigrid Method: Theory, Application and Supercomputing. New York, 1988
65 曹志浩编著. 多格子方法. 上海：复旦大学出版社，1987
66 Wesseling P. An Introduction to Multigrid Methods. New York: John Wiley & sons, 1992
67 张培强主编. Matlab 语言——演算纸式的科学工程计算语言. 合肥：中国科学技术大学出版社，1995
68 肖劲松主编. Matlab 5. X 与科学计算. 北京：清华大学出版社，2000

数学科学的成就已成为当今高科技时代进步发展的重要基础，应用数学的发展是科技工业兴旺发达的强有力支柱。为了迎接21世纪的挑战，本手册向您介绍现代应用数学的各个分支，为您在解决科研、教学以及生产实践的各种问题中，提供不可缺少的工具。全书共计六卷：

《运筹学与最优化理论卷》
《现代应用分析卷》
《概率统计与随机过程卷》
《离散数学卷》
《分析与方程卷》
《计算与数值分析卷》

——各卷内容自成体系，互相独立，方便读者按需选用。

ISBN 7-302-09831-X



定价：48.00元